

AD-A068 890

HARRIS CORP MELBOURNE FLA

F/G 17/4

APPLICATION OF A CORRELATION DISCRIMINANT OPERATOR TO PERTURBAT--ETC(U)

APR 79 G P MARTIN

F30602-77-C-0073

NL

UNCLASSIFIED

RADC-TR-79-44

1 OF 4
ADA
068890



DDC FILE COPY

ADA068890

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM																
1. REPORT NUMBER RADC TR-79-44	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER																
4. TITLE (and Subtitle) APPLICATION OF A CORRELATION DISCRIMINANT OPERATOR TO PERTURBATIONAL ADAPTIVE ALGORITHMS		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report May 1977 - May 1978																
6. AUTHOR(s) G. Patrick Martin		7. PERFORMING ORG. REPORT NUMBER N/A																
8. PERFORMING ORGANIZATION NAME AND ADDRESS Harris Corporation P.O. Box 37 Melbourne FL 32901		9. CONTRACT OR GRANT NUMBER(s) F30602-77-C-0073 new																
10. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (DCID) Griffiss AFB NY 13441		11. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 5556																
12. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same		13. REPORT DATE Apr 1979																
14. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		15. NUMBER OF PAGES 308																
16. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same		17. SECURITY CLASS. (of this report) UNCLASSIFIED																
18. SUPPLEMENTARY NOTES RADC Project Engineer: Peter Edraos (DCID)		18a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A																
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)																		
<table border="0"> <tr> <td>Adaptive Algorithm</td> <td>Antenna Nulling</td> <td>Correlators</td> </tr> <tr> <td>Adaptive Antenna</td> <td>Antenna Steering</td> <td>Digital Search</td> </tr> <tr> <td>Adaptive Array</td> <td>Antijamming</td> <td>ECCM</td> </tr> <tr> <td>Adaptive Process</td> <td>Beam Steering</td> <td>Random Search</td> </tr> <tr> <td>Adaptive Processor</td> <td>Correlation</td> <td>Signal Correlation</td> </tr> </table>				Adaptive Algorithm	Antenna Nulling	Correlators	Adaptive Antenna	Antenna Steering	Digital Search	Adaptive Array	Antijamming	ECCM	Adaptive Process	Beam Steering	Random Search	Adaptive Processor	Correlation	Signal Correlation
Adaptive Algorithm	Antenna Nulling	Correlators																
Adaptive Antenna	Antenna Steering	Digital Search																
Adaptive Array	Antijamming	ECCM																
Adaptive Process	Beam Steering	Random Search																
Adaptive Processor	Correlation	Signal Correlation																
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)																		
<p>A primary objective of this study was to develop and apply a newly conceived, generalized technique for adaptive array reference function generation to a large class of systems having no null steering AJ protection. Because the reference function generation technique, referred to here as a correlation discriminant operator or signal recognizer, does not depend upon filtering unwanted noise, it can be orders of magnitude faster than conventional reference function generation approaches. Thus a correlation discriminant operator.</p> <p style="text-align: right;">(Cont'd)</p>																		

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

391655

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Item 20 (Cont'd)

facilitates the realization of a fast adaptive signal-to-noise ratio maximizing algorithm even in the case of narrowband desired signals having narrowband a priori discriminants. In addition, the correlation discriminant operator enables an algorithm to achieve optimum expected value adaptation transients and steady-state solutions.

Another primary objective was to conceive, develop and simulate a technique for utilizing existing high-quality communication receivers as integral components in a high performance, null steering AJ subsystem. The thrust of this task was to provide null steering AJ to a minimally modified communications network while avoiding replacement of expensive high-quality radios. As is shown in the report, these objectives can be achieved with the result, according to simulation, being a practical, low cost null steering AJ subsystem.

In order to inherently utilize a communications receiver within the null steering processor, a perturbational algorithm was developed. This algorithm incorporates the new reference function generation technique to eliminate the conventional performance measure device with its time delay and desired signal estimation bias. In this form, the algorithm is suitable for use in slightly modified existing communications systems, enabling the use of existing communication sets; the algorithm is realized in a microprocessor which requires samples of the communications receiver's IF output. Simulation results predict an S/N improvement of 40 dB in under 20 ms using a generic HF/VHF/UHF receiver.

In addition, an expected value equivalence discovered between the conventional random search algorithm and normalized gradient algorithm is reported. Also presented is a generalized algorithm for the case of optimum reception of several simultaneous desired signals in the presence of jamming.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

ACKNOWLEDGEMENTS

The author is indebted to George Olive who was responsible for most of the computer simulations reported here and for several important suggestions regarding the algorithms. Credit is also due to Richard Bustelo, Dr. David Bell and Larry Hand for their microprocessor development and programming work.

Special thanks are due to Dr. Jack Henry for his thought provoking comments regarding realization of perturbational algorithms and to Dr. Charles Zahm for analyzing the bandpass filter signal recognizer. Appreciation is extended as well to Dr. George Rassweiler and Dick Davis for numerous helpful suggestions and comments.

Regarding manuscript preparation, special notice must be given to Melodie Poe for her exceptional work. Additional thanks are due to Ruth Forsman, Sue Hestbeck, and Raymond Andrews.

ACCESSION for	
NTIS	Write Section <input checked="" type="checkbox"/>
DDC	Bull Section <input type="checkbox"/>
UNANNOUNCED	
JUL 1 1984	
BY	
DISTRIBUTION/ADMT/ADMT CODES	SP. CIAL
A	

TABLE OF CONTENTS

<u>Paragraph</u>	<u>Title</u>	<u>Page</u>
1.0	INTRODUCTION	1
1.1	General Comments	1
1.2	Report Organization	6
2.0	CORRELATION DISCRIMINANT OPERATOR	10
2.1	Conceptual Description	11
2.1.1	CDO Use in an Adaptive Array	13
2.1.2	CDO Use in a Modem	19
2.2	Theory	21
2.2.1	Signal Recognizer Theory	21
2.2.2	Signal Recognition Using a Bandpass Filter	28
2.2.3	Signal Recognition Using a Time Domain Filter	36
2.3	Signal Recognizer Examples	38
2.3.1	PSK Signal Recognizer	38
2.3.2	AM Signal Recognizer Example	42
2.3.2.1	Frequency Domain Interpretation	44
2.3.2.2	Comparison with a Customary Approach	49
2.3.3	A Chirp Signal Recognizer	51
2.3.4	FSK Signal Recognizer Example	54
2.3.5	FM Signal Recognizer Circuit	56
3.0	ALGORITHMS	59
3.1	Basic Differential Equations	61
3.1.1	Basic Positive Signal Feedback Algorithm	61
3.1.2	Filtered Steering Vector PSF	68
3.1.3	Expanded PSF Algorithm	73
3.2	Perturbational Correlation	76

Continued

<u>Paragraph</u>	<u>Title</u>	<u>Page</u>
3.2.1	Gradient Measurement	78
3.2.1.1	With Conventional Performance Measure	78
3.2.1.2	With CDO and Multipliers	89
3.2.2	Perturbational Sequences	93
3.2.2.1	Random Sequences	93
3.2.2.2	Pseudo-random Sequences	93
3.2.2.3	Orthogonal Sequences	94
3.2.2.3.1	Three Level Sequence	94
3.2.2.3.2	Walsh Functions	95
3.2.2.4	FDMA Sequences	99
3.3	Modified Algorithms	99
3.3.1	Perturbational PSF Algorithm	102
3.3.2	FDMA of Search Parameters	104
3.3.3	Random Search	107
3.3.4	Log Power Algorithm	115
4.0	AJ SYSTEM DESIGN	120
4.1	Postulated System Problem	121
4.1.1	Low Cost/High Quality Null Steering Anti-Jam	121
4.1.2	Complementary Spread Spectrum Anti-Jam Capability	121
4.1.3	Existing Communication Sets	122
4.1.4	Compatibility	123
4.1.5	Present System Features	123
4.2	Selected Approach	124
4.2.1	Selected Algorithm	124
4.2.2	Selected Discriminant Function	125
4.3	Detailed Design	133
4.3.1	Antenna	137
4.3.2	Communications Set Modifications	141
4.3.3	Adaptive Processor	144
4.3.3.1	Algorithm	144
4.3.3.2	Perturbational Sequences	144

Continued

<u>Paragraph</u>	<u>Title</u>	<u>Page</u>
4.3.3.3	IF Output Sampling	145
4.3.3.4	SR Realization	147
4.3.3.5	AGC	150
4.3.3.6	Microprocessor	151
4.3.3.6.1	General Design Features	151
4.3.3.6.2	Demonstration Unit Design	154
4.3.3.6.2.1	Firmware Description	157
4.3.3.6.2.2	Software Description	160
4.3.4	Modem	160
4.3.5	Discriminant Function Considerations	164
4.3.5.1	Multiplicative Sequence	164
4.3.5.1.1	Slow Sequence	165
4.3.5.1.2	Fast Sequence Slow Sequence Product	166
4.3.5.2	Additive Code	167
4.3.5.3	Carrier Alone	169
4.3.5.4	Uncoded Pilot	171
5.0	SIMULATION/EXPERIMENT	172
5.1	Simulation Methods	173
5.2	Microprocessor Experiments	179
5.2.1	Intel 8080A Experiment	179
5.2.1.1	Hardware	180
5.2.1.2	Algorithms	181
5.2.1.3	Results	182
5.2.2	AMD 2900 Experiment	184
5.2.2.1	Hardware	184
5.2.2.2	Algorithm	185
5.2.2.3	Results	185
5.2.3	Fortran Microprocessor Experiment	186
5.2.3.1	Hardware	187
5.2.3.2	Algorithms	188

Continued

<u>Paragraph</u>	<u>Title</u>	<u>Page</u>
5.2.3.3	Results	188
5.3	Simulation Results	189
5.3.1	Standard Case	190
5.3.2	Array Gain	201
5.3.3	Perturbational Sequence Effects	201
5.3.3.1	Perturbation Amplitude	204
5.3.3.2	Step-by-Step	210
5.3.3.3	Continual/Periodic Weight Update	210
5.3.3.4	Perturbation Sequence Length	219
5.3.4	Alpha Loop Gain	224
5.3.5	Signal Recognizer Parameters	228
5.3.6	AGC Effects	240
5.3.7	Digital Word Size	240
5.3.7.1	No AGC	240
5.3.7.2	With AGC	245
5.3.8	Rotation Effects	251
5.3.9	Two Jammers	258
5.3.10	Instabilities	262
5.3.10.1	Array Gain Instability	262
5.3.10.2	Receiver Time Delay	265
5.3.11	Signal Power	273
6.0	CONCLUSIONS AND RECOMMENDATIONS	276
6.1	Conclusions	276
6.2	Recommendations	277
6.2.1	Direct Applications	278
6.2.2	Related Topics	279
6.2.3	New Directions	280
APPENDIX A	GENERAL CALCULATION OF PERTURBATIONAL ALGORITHM SIGNAL RECOGNIZER OUTPUT	282
APPENDIX B	AGREEMENT OF COMPUTER SIMULATIONS	286
APPENDIX C	LISTING OF AMD 2900 MICROPROCESSOR PROGRAM	290

EVALUATION

This study directly attacks a present critical need in the area of electronic signal processing for antijam systems - the need for a satisfactory method of discriminating between wanted and unwanted signals. This area of investigation applies to RADC TPO 4A: Communications ECCM. The key point of this study is the recognition of a simple concept, based upon interference decorrelation rather than signal correlation processes. The study has been involved with determination of the validity of the concept, the feasibility of applying this concept to ECCM communication systems, and the various methods by which this concept can be applied.

Peter N. Edraos
PETER N. EDRAOS
Project Engineer

1.0 INTRODUCTION

A primary objective of this study was to develop and apply a newly developed, generalized technique for adaptive array reference function generation to a large class of systems having no null steering antijam protection. Because the reference function generation technique, referred to here as a correlation discriminant operator or signal recognizer, does not depend upon filtering unwanted noise, it can be orders of magnitude faster than conventional reference function generation approaches. Thus, a correlation discriminant operator facilitates the realization of a fast adaptive signal-to-noise ratio maximizing algorithm even in the case of narrowband desired signals having narrowband a priori discriminants. In addition the correlation discriminant operator enables an algorithm to achieve optimum expected value adaptation transients and steady-state solutions.

Another primary objective was to conceive, develop and simulate a technique for utilizing existing high-quality communication receivers as integral components in a high performance, null steering AJ subsystem. The thrust of this task was to provide null steering AJ to a minimally modified communications network while avoiding replacement of expensive high-quality radios.

As we shall show in the following chapters, these objectives can be achieved with the result, according to simulation, being a practical, low cost, high performance null steering AJ subsystem. Substantial development of both correlation discriminant operator and perturbational algorithm theory was required in order to meet these objectives.

1.1 General Comments

In the following paragraphs, we will review a conventional temporal reference perturbational technique and identify two important problems thereof. We then indicate how these problems can be overcome, details of which form the basis of this report.

Most high performance, state-of-the-art adaptive arrays employ RF or IF correlation to extract the gradient of the performance

measured to be maximized. In order to restrict the correlator, and thus the adaptive processor, to the RF bandwidth of interest and to provide adequately high level RF/IF inputs, one needs in effect the RF and IF stages of a high quality receiver at each antenna input. While this approach in itself is expensive, it is also economically important that there are many thousands of currently unprotected radio links which need AJ protection and which already have high quality expensive radios.

There are several ways an algorithm can be utilized to minimize the need for expensive RF/IF correlators. One can eliminate these components altogether with a method using weight perturbations coupled with measurement of receiver output performance measure changes. We will refer to such a technique as a perturbational algorithm. Random search optimization is an example of such a perturbational approach. As we will show in Chapter 3.0, weight perturbations serve to provide a means for performance measure gradient extraction.

A representative temporal reference conventional adaptive array/receiver circuit employing a perturbational approach is illustrated in Figure 1.1-1. At the top of the figure, a single weighted antenna input of several is illustrated. The induced voltage from this antenna element, referred to as $x_i(t)$, is multiplied by a weight and perturbation value, respectively $[W_i + p_i(t)]$ and added to the other weighted inputs by an array summer. The output of the summer is directed to a receiver which provides gain and filtering to the bandwidth of interest. The receiver's IF output is directed to circuits which respectively approximately calculate the total output power and output signal power. These measurements are input to a device which performs a relative power comparison, the performance measure. Adjustment of the array weights proceeds on the basis of whether or not the performance measure is improved as different values of weights and perturbations are applied.

As shown, array output signal plus noise power is calculated by squaring the receiver's IF output voltage and lowpass filtering this quantity. In a conventional manner, the signal output power detector

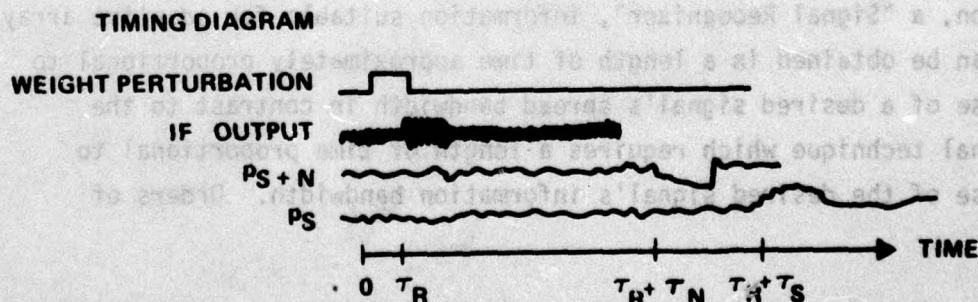
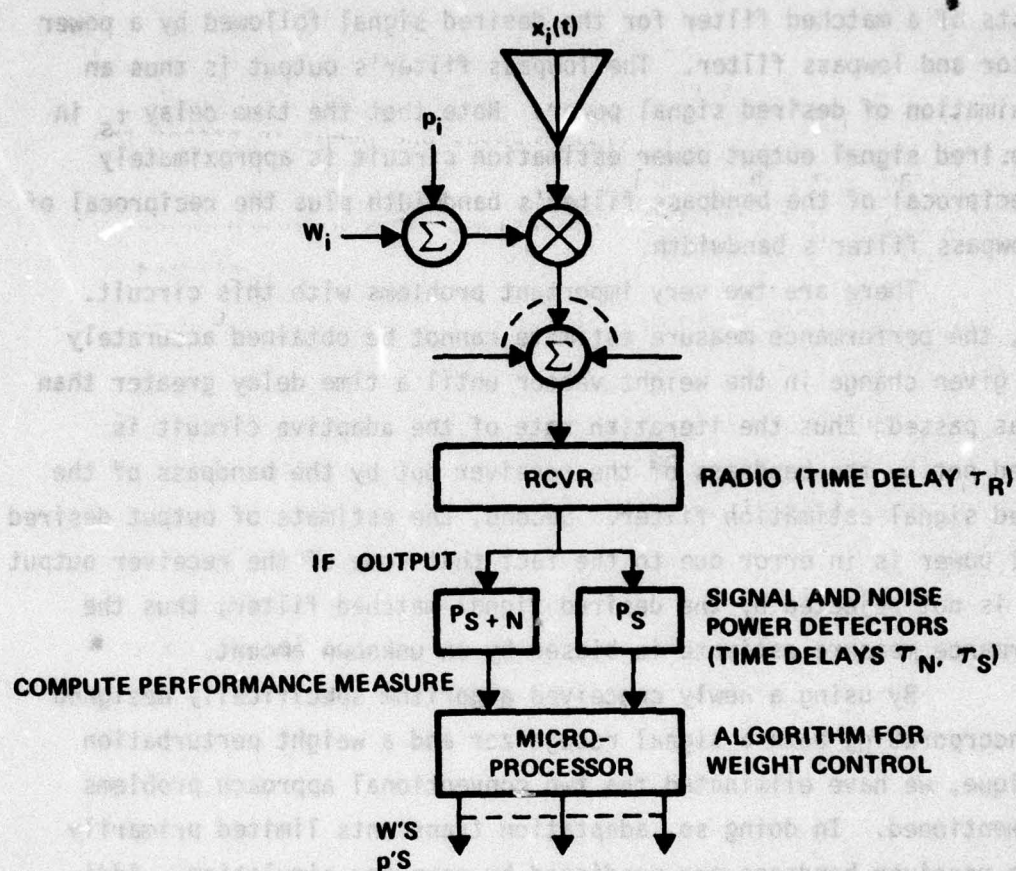


Figure 1.1-1. Representative Conventional Temporal Reference Perturbational Adaptive Array/Receiver

consists of a matched filter for the desired signal followed by a power detector and lowpass filter. The lowpass filter's output is thus an approximation of desired signal power. Note that the time delay τ_s in the desired signal output power estimation circuit is approximately the reciprocal of the bandpass filter's bandwidth plus the reciprocal of the lowpass filter's bandwidth.

There are two very important problems with this circuit. First, the performance measure estimate cannot be obtained accurately for a given change in the weight vector until a time delay greater than τ_s has passed; thus the iteration rate of the adaptive circuit is limited not by the bandpass of the receiver but by the bandpass of the desired signal estimation filter. Second, the estimate of output desired signal power is in error due to the fact that some of the receiver output noise is not rejected by the desired signal matched filter; thus the performance measure estimate is biased by an unknown amount.

By using a newly conceived algorithm specifically designed for incorporating both a signal recognizer and a weight perturbation technique, we have eliminated the two conventional approach problems just mentioned. In doing so, adaptation transients limited primarily by the receiver bandpass are predicted by computer simulation. Additionally, optimum steady-state solutions are predicted. Key factors in the new development are as follows.

Primarily, emphasis is shifted from the desired signal estimation function to a discrimination function. In the case of spread spectrum, use of a correlation discriminant operator, or its practical realization, a "Signal Recognizer", information suitable for adaptive array control can be obtained in a length of time approximately proportional to the inverse of a desired signal's spread bandwidth in contrast to the conventional technique which requires a length of time proportional to the inverse of the desired signal's information bandwidth. Orders of

magnitude improvement in the time delay reduction can therefore be obtained with corresponding improvement of adaptive array response time.

Even if the desired signals are not spread spectrum and instead are identified through their carrier or a pilot tone, substantial improvements continue to be obtained through use of the correlation discriminant operator because conventional techniques would narrowband filter either the carrier or the pilot while such filtering is not required through use of the operator function.

Another key factor is the conception and development of perturbational algorithms specifically designed for the incorporation of a signal recognizer and permitting the realization of adaptive algorithms in their usual differential equation form such as with direct i.e., RF/IF, correlation. This approach is contrasted to the conventional perturbational approach illustrated earlier in Figure 1.1-1.

By way of performance comparison, if the receiver is a typical HF/VHF/UHF radio, and if the desired signal a priori discriminant is the carrier or a slowly varying pseudo random sequence, the time required to achieve a S/N improvement of 8.7 dB using the conventional approach is in the order of 100 milliseconds; alternatively, the newly devised approach is capable of achieving the same performance in about one millisecond. This simulated performance is as good as that obtained by some high quality direct (RF/IF) correlation algorithms.

We note that large arrays having a main beam constraint can effectively use desired signal direction of arrival information even though this parameter may be somewhat inaccurate. Such an array may be faster than a conventional temporal reference array because a signal power detector is not needed. The above comparisons, therefore, apply to a relatively small array wherein "main beam" constraints are meaningless.

In the course of meeting our primary objectives, some potentially useful new algorithms were created and existing ones, such as the Positive Signal Feedback or PSF, were substantially improved. Particularly, a significant expansion of the PSF algorithm was conceived which allows the PSF to simultaneously maximize several desired signals. Such a technique is useful in the system concept of conferencing users or in a multiplexed

system where gain must be maintained in several directions so as to permit rapid random access for those users.

While investigating perturbational algorithms in a general sense, we were able to show an expected value equivalence between the conventional random search algorithm and a normalized gradient algorithm. We believe this to be an important result, and it is expected that in future work this equivalence will lead to better understanding and thus better design of search algorithms.

In the experimental portion of our work, we designed, developed, and simulation tested two different microprocessor controllers which exercised several of the new S/N maximizing perturbational algorithms. This testing was facilitated by connecting the microprocessors to a general-purpose computer which was programmed to simulate the electromagnetic environment/antenna array/receiver/analog-digital interface.

The simulation results obtained are encouraging in that they verify the ability of the modified PSF algorithm using a signal recognizer to achieve adaptation speeds limited primarily by the receiver bandwidth and not by the performance measure estimation device as is the case in conventional designs.

Although this study was directed primarily toward narrowband radio applications, the results are general and could be used to substantially reduce the cost and improve the performance of future adaptive array AJ systems.

1.2 Report Organization

In Chapter 2.0, we present a comprehensive discussion of the correlation discriminant operator (CDO) and its practical realization, the Signal Recognizer. The CDO is the key element in the development of our improved perturbational algorithms. This chapter begins with a clarification of the concepts of desired signal estimation versus desired signal discrimination capability. In extracting the information content of a desired signal, it is necessary to produce as good an estimate of the desired signal as is feasible. In the case of an

adaptive processor, however, the necessary "information" is a desired signal steering vector which varies much more slowly than the signal's modulation. In this case, a sufficient function for control of the adaptive array weights can be obtained through a discrimination technique based on an implicit decorrelation of desired signal and noise and interference. This discrimination operation can be much faster than the signal estimation function. Following such introductory comments, we present the general equations of the correlation discriminant operator. Numerous examples then follow including the cases of direct spread pseudo noise, ordinary AM, FSK, chirp (parabolic phase modulation), and narrowband FM. A distinction is made between wideband and narrowband CDO devices and the impact of a resulting phase term is discussed.

Chapter 3.0 is devoted to a presentation of the algorithms devised and studied in the course of this work. All of these algorithms are designed to maximize S/N ratio in a broadband sense. This chapter begins with a general discussion concentrating on the Positive Signal Feedback algorithm and its modifications. An important generalization of the selected PSF algorithm for multiple simultaneous desired signals is also described. This technique is designed to simultaneously maximize the product of S/N ratio from independent desired signal sources (such as during system conferencing).

Following this general presentation is a comprehensive discussion of how the general algorithms are utilized in a perturbational context; this of course is a specific interest of this study. We show that the perturbational technique in its many variations is strictly a way of obtaining an estimate of the error surface gradient, even though this fact is sometimes implicitly concealed. These derivations lead naturally to a presentation of the expected value equivalence of random search and gradient algorithms which was discovered during the course of this study. Following the presentation of this equivalence is a comparison of conventional perturbational algorithms with those incorporating a correlation discriminant operator. Here we show how

orders of magnitude improvement in ultimate adaptation rate can be achieved by using the operator function. Finally, a variation of the perturbational algorithms using a specially modified receiver is presented wherein the perturbational terms do not contribute to a degradation in the receiver's output S/N ratio.

Results from Chapters 2.0 and 3.0 are brought together in Chapter 4.0 to produce an example AJ system design. This chapter begins with a description of the postulated scenario; an existing narrowband radio communications network needing adaptive array AJ capability. Next, a selected approach is detailed. Basically, the approach uses existing receivers to produce a low-cost design that is compatible with minimally modified radios also present in the communications network. Three classes of users may be present, each receiving various degrees of maximum AJ capability. Following this general system description is a detailed design of the adaptive array processor including such topics as Algorithm Description, Receiver Modifications, Microprocessor Design, Perturbational Sequences, AGC, A/D Conversion and Sampling, Antenna Configuration, and Size, Weight and Power Estimates.

Verification of the adaptive array AJ system design given in Chapter 4.0 was achieved through a simulation of the electromagnetic environment/antenna/array/receiver which was interfaced with either an actual microprocessor or a simulation of the microprocessor. Chapter 5.0 begins with a detailing of this simulation. Here, we discuss strengths and weaknesses of the simulation, its general capabilities as well as approximate restrictive versions which were used to obtain quick estimates of performance explained herein. Much of our presented simulation data are based upon a total simulation of the AJ subsystem; that is, the actual microprocessor was also simulated. In order to verify accuracy of the total simulation, we present computer runs of the actual microprocessor adaptation transients which agree with the total simulation adaptation transients.

The next part of Chapter 5.0 is a description of the microprocessor adaptive array controller experiment. We begin by describing

the experimental configuration, specifically the connection of the simulation computer to the actual microprocessors, and include details of the total simulation.

Following this description of the experimental hardware, simulation results are presented. All important algorithm parameters were varied and representative results are given. The data of this section are therefore useful not only for demonstrating the effect of specific parametric variations but for obtaining engineering design factors as well. Specific identification of the parameters investigated is given in the introduction of Section 5.3.

Finally, in Chapter 6.0, we summarize the findings of this study and make recommendations for the applications of this new technology.

2.0 CORRELATION DISCRIMINANT OPERATOR

This chapter details a generalized practical technique for obtaining a "desired signal reference function" as required for optimal performance of a temporal reference S/N ratio maximizing adaptive array. Specifically, waveforms containing the desired signal, noise and interference are subjected to an operator function referred to herein as a "correlation discriminant operator" which yields precisely in the context of the algorithm's average value adaptation transient and optimum solution an ideal desired signal reference function.

A correlation discriminant operator requires an a priori discriminant which uniquely identifies the desired signal. Specifically, a desired signal must be composed of products, a suitable term of which must be known. This requirement is contrasted to the customary formulation which requires that the entire signal be known.

In the following sections of this chapter, we will discuss the ideal Correlation Discriminant Operator (CDO) and show how it is capable of enabling an adaptive algorithm to achieve optimum expected value adaptation transients and solutions. In Section 2.2, we introduce the "signal recognizer" which we define to be a practical realization of the ideal CDO. This section is divided into three parts. The first, Section 2.2.1, is devoted to signal recognizers realized with a time delay while Section 2.2.2 is an analysis based on substitution of a bandpass filter for the time delay. A time domain analog of the frequency domain bandpass filter analysis is briefly given in the following section, 2.2.3.

Additional theoretical analysis of a practical nature is given in Section 2.3 which is devoted to specific signal recognizer examples for several waveforms of interest. This section is more than simply a guide to the practical use of the CDO in that the examples provide considerable insight into the principle of operation.

2.1 Conceptual Description

A correlation discriminant operator transforms a function having both desired and undesired terms so as to cause the output desired terms to largely retain their correlation properties and to simultaneously cause the output undesired terms to become uncorrelated with any other waveform present in the system.

A required input to the CDO is an a priori known product term of the desired waveform. For example, consider the desired signal $S(t)$ where

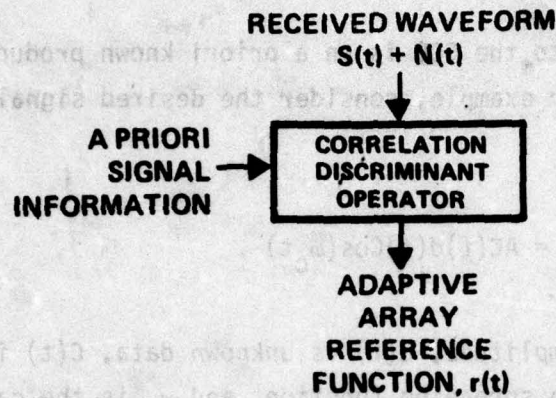
$$S(t) = AC(t)d(t)\cos(\omega_c t) . \quad 2-1$$

In the above, A is unknown amplitude, $d(t)$ is unknown data, $C(t)$ is a known pseudo-random bandwidth spreading function, and ω_c is the carrier frequency which may be doppler shifted by an unknown and arbitrary amount. As we will show in Section 2.3, either $C(t)$ or $\cos(\omega_c t)$ can be chosen as the a priori input.

A conceptual CDO block diagram is given in Figure 2.1-1. In this example, a received waveform consisting of signal and noise is input to the CDO as is a priori signal information. Output of the CDO is the function $r(t)$ which consists of signal and an uncorrelated noise $n(t)$. The required zero correlation with $n(t)$ is expressed as follows. We define

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} S(t)n(t)dt = 0 . \quad 2-2$$

Also, we define



$$r(t) = S(t) + \eta(t)$$

WHERE: $0 = \frac{1}{T} \int_t^{t+T} S(t) \eta(t - \tau) dt$

$$0 = \frac{1}{T} \int_t^{t+T} N(t) \eta(t - \tau) dt$$

TRUE FOR ALL τ

Figure 2.1-1. Conceptual Correlation Discriminant Operator

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} n(t)n(t)dt = 0 . \quad 2-3$$

In some applications, a signal recognizer realization of the CDO has an output signal term which is only partially correlated with the input signal term. In this case, we can show that the uncorrelated portion of $S(t)$ can be incorporated into $n(t)$ without loss of generality. Although output desired signal is not a direct replica of the input desired signal, it is nevertheless highly correlated with the input signal, and from a correlation standpoint the "lost" signal power can be made up with amplifier gain.

Before proceeding with a detailed theoretical treatment, it is helpful to understand how the CDO will be used. Although it has other applications, the principal use envisioned is for the generation of the desired signal input of a temporal reference adaptive array for the maximization of S/N ratio.

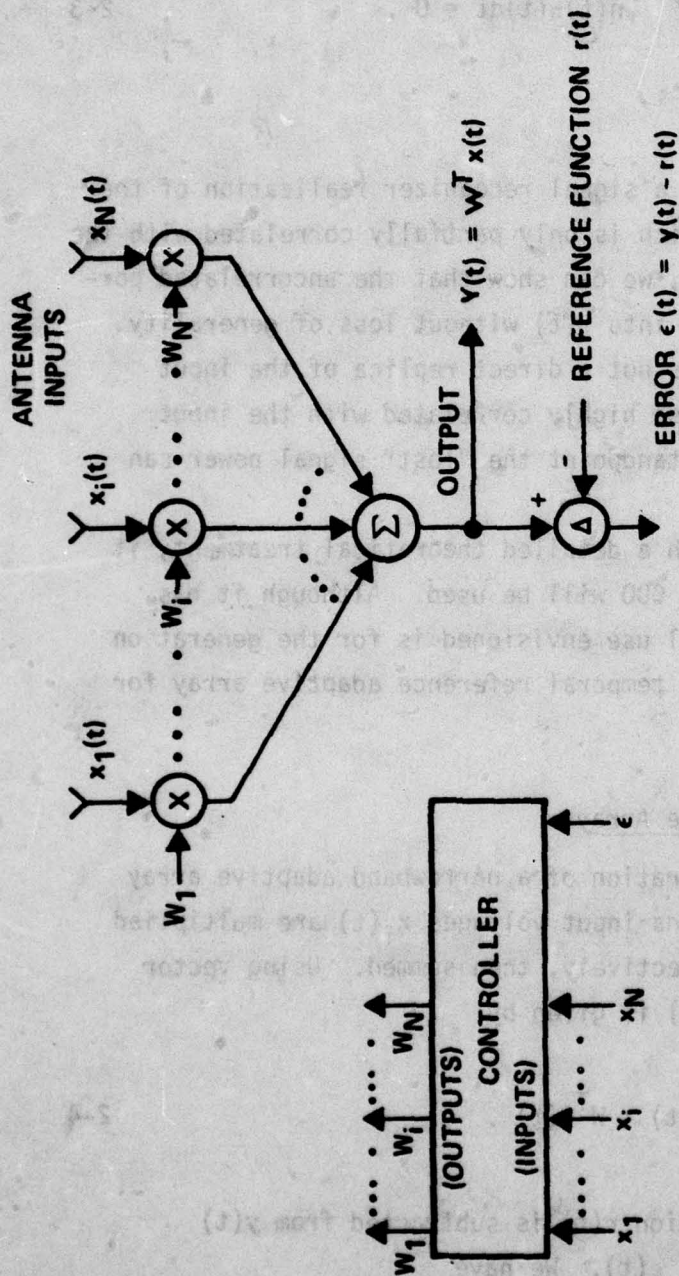
2.1.1 CDO Use in an Adaptive Array

A generalized configuration of a narrowband adaptive array is given in Figure 2.1-2. Antenna input voltages $x_i(t)$ are multiplied by the adaptive weights W_i , respectively, then summed. Using vector notation, the output voltage $y(t)$ is given by

$$y(t) = W^T x(t) . \quad 2-4$$

A desired signal reference function $r(t)$ is subtracted from $y(t)$ yielding the array error voltage $\epsilon(t)$. We have

$$\epsilon(t) = y(t) - r(t) . \quad 2-5$$



TASK: MINIMIZE $\epsilon(t)$ IN AN AVERAGE POWER SENSE (MEAN SQUARE ERROR CRITERIA)

$$\overline{\epsilon^2} = \overline{(y - r)^2}$$

CLASSICAL PROBLEM: $r(t)$ IS UNAVAILABLE

Figure 2.1-2. Narrowband Reference Adaptive Array

We assume that $x(t)$ is the sum of desired signal $S(t)$ plus undesired noise $n(t)$ which may consist of both thermal noise and interference waveforms.

In this example, we will follow convention and assume that the algorithm is designed to minimize the expected value of the error power in a mean-square sense. We begin by calculating the expected value of the error power. One gets

$$E\{\epsilon^2\} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \epsilon^2(t) dt . \quad 2-6$$

A minimum of this function can be found by setting the gradient with respect to the adaptive weights to zero. We get

$$\nabla_w E\{\epsilon^2\} = 0 . \quad 2-7$$

Where the symbol ∇_w is defined as follows:

$$\nabla_w = a_1 \frac{\partial}{\partial w_1} + a_2 \frac{\partial}{\partial w_2} + \dots + a_n \frac{\partial}{\partial w_n} . \quad 2-8$$

At this point, we invoke the usual quasi-static assumption which allows us to treat the adaptive weight W as though it were static. When the quantity $x(t)$ is an RF voltage and the array adaptation time is relatively long compared with a reciprocal RF bandwidth, then this is a very reasonable approximation. Given a very fast adapting array and narrowband RF quantities, this assumption may not be valid.

Through use of the quasi-static approximation, the gradient operation and expected value operation may be interchanged. We write

$$E\{\nabla_w \epsilon^2\} = 0 = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \nabla_w \epsilon^2(t) dt . \quad 2-9$$

Expanding (2-7) further yields

$$E\{\nabla_w \epsilon^2\} = 2E\{\epsilon \nabla_w \epsilon\} . \quad 2-10$$

Finally, substituting (2-5) and evaluating the components of (2-10) yields

$$2E\{\epsilon \nabla_w \epsilon\} = 2E\{xx^T W - xr\} = 2[R_x W - E\{xr\}] . \quad 2-11$$

Where we have made the following customary definition of the cross correlation matrix R_x

$$R_x = E\{x(t)x^T(t)\} . \quad 2-12$$

The weight vector which solves Equation (2-9) may be calculated by applying the inverse of R_x to (2-11). We get

$$W_{opt} = R_x^{-1} E\{xr\} . \quad 2-13$$

In an ideal adaptive array, the reference function $r(t)$ is assumed to be an a priori known exact copy of the desired signal. Referring to this ideal a priori known desired signal as $S_o(t)$ we may evaluate (2-13) for the optimum weights. We get

$$W_{opt} = R_x^{-1} E\{S(t)S_o(t)\} . \quad 2-14$$

Where we have used the assumption that

$$E\{n(t)S_0(t)\} = 0 .$$

2-15

That is, we have assumed that the signal is uncorrelated with any of the noise present in the system.

Observe that (2-14) is a vector equation since $S(t)$ is the vector of input signals and the desired signal term $S_0(t)$ is a scalar quantity. The expected value term in (2-14) is referred to as the desired signal steering vector.

The customary problem associated with this temporal reference type of adaptive array has been that the reference function $r(t)$ is difficult to obtain. It has been often mentioned that if one knew the reference function as well as one needed to know it, then there would be little need for the array. While some investigators have sought to obtain an approximation to the desired signal reference term $S_0(t)$ by filtering signal plus noise waveforms, we address this problem through use of the correlation discriminant operator.

We can obtain the term $[S_0(t) + n(t)]$ from a variety of sources. For example, it can be obtained directly from one of the antenna inputs. Assuming no dispersion in the propagation path, such a signal would contain an exact replica of the desired waveform. Alternatively, the desired signal plus noise waveform can be obtained from the array output. The advantage of obtaining the signal from this point is that undesired noise is reduced as the array adapts.

Submitting signal plus noise to a correlation discriminant operator, as shown in Figure 2.1-1, yields

$$r(t) = S_0(t) + n(t) . \quad 2-16$$

When (2-16) is substituted into (2-13), one may again calculate an expected value weight solution. Expanding the xr product gives

$$W = R_x^{-1} E\{S(t)S_o(t) + n(t)S_o(t) + S(t)n(t) + n(t)n(t)\} . \quad 2-17$$

As before, signal and noise $n(t)$ are uncorrelated. Furthermore, according to the definition of a correlation discriminant operator, we also have that

$$E\{s(t)n(t)\} = 0 . \quad 2-18$$

and that

$$E\{n(t)n(t)\} = 0 \quad 2-19$$

thus, the resulting expected value weight vector is

$$W = R_x^{-1} E\{S(t)S_o(t)\} = W_{opt} . \quad 2-20$$

The conclusion of this derivation is that through the use of a correlation discriminant operator to generate the adaptive array reference function, an optimum weight vector solution is obtained in an expected value sense just as though one had available the a priori known ideal desired signal reference function.

We will show later that use of the correlation discriminant operator in an array having a finite adaptation time does result in weight jitter above that which would be seen in the ideal array. By suitable algorithm design (see Chapter 3.0), this weight jitter term can be made negligibly small in many applications. It is furthermore noted that even with an exactly known ideal reference function (noise term $n=0$) an array with a finite adaptation time will always display

- * weight jitter due to other causes such as cross correlation of desired signal and interference waveforms.

When specific examples of the CDO are given in 2.3, the reader may note that some adaptive array designers have been inadvertently using a CDO. In such cases, their adaptive array could achieve optimum results even though the investigator may have expected only approximately optimum solutions. The derivations to be presented in the remainder of this chapter, therefore, may be used to evaluate previous approaches and explain why either better than expected or worst than expected results have been obtained when using intuitively designed reference function generation circuits.

2.1.2 CDO Use in a Modem

Although our principal interest here is the adaptive array application, we are providing this example to illustrate the utility of the CDO in an application which requires either that a desired signal's power be measured without a bias or alternatively, if the desired signal consists of a modulated carrier, that the carrier's second harmonic be generated without a noise bias. Reference to Figure 2.1-3 illustrates this operation.

A signal plus noise input is divided into two paths, one of which leads to a CDO and then to a multiplier while the other path leads directly to the multiplier. According to CDO theory, the multiplier output is

$$v(t) = S^2(t) + S(t)n(t) + S(t)n(t) + n(t)n(t) \quad 2-21$$

For $S(t)$ as given by (2-1), $S^2(t) = \frac{A^2}{2} [1 + \cos 2\omega_c t]$. It is evident that the expected value of Equation (2-21) is signal power. It may not be evident until Section 2.3 is reviewed, but the second harmonic of $S^2(t)$ is also unbiased by the remaining terms in (2-21) due to the lack of correlation between the system noise $n(t)$ and the CDO noise $n(t)$. Thus, a phase lock loop tuned to approximately $2\omega_c$ can lock without bias.

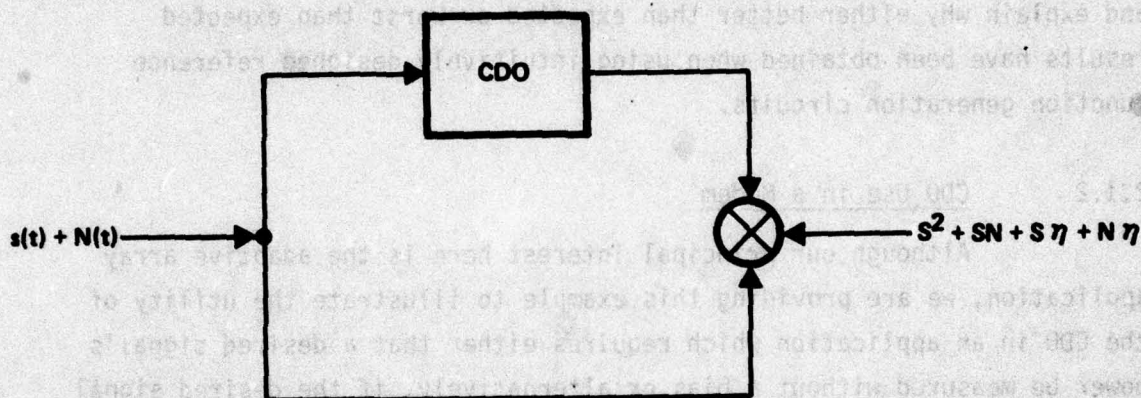


Figure 2.1-3. Use of a Correlation Discriminator Operator to Obtain an Unbiased Measurement of $s^2(t)$

This circuit can be redrawn for a more practical implementation using the CDO designed for an AM desired signal modulation. In that example, it is assumed that the carrier frequency is the a priori known discriminant. Since it is carrier frequency we wish to track in this application, an amplitude modulation CDO can be ideally applied.

2.2 Theory

In the previous section, we discussed the properties of an ideal correlation discriminant operator without regard to physical realizability. In this section, we present an analysis of a practical circuit which essentially realizes the necessary features of the CDO. We will refer to this circuit as a "Signal Recognizer." Section 2.2.1 is devoted to analysis of a signal recognizer circuit which contains the minimum number of essential components. This circuit, realized with a time delay, also has major practical application potential. In Section 2.2.2, a signal recognizer incorporating a bandpass filter for realizing the necessary time delay as well as for providing reduction in the noise component $n(t)$ is analyzed.

2.2.1 Signal Recognizer Theory

Two practical Signal Recognizer circuits are illustrated in Figure 2.2-1. As shown in the figure, the signal recognizer consists of a multiplication by the a priori known function, delay in time by the amount τ , and multiplication again by the identical a priori known function. Equivalently, one can delay the signal plus noise by an amount τ and then multiply by the combined product of a priori known function times delayed a priori known function. In a preview of later derivations, the time delay τ will be chosen so that the a priori known function times the delayed a priori known function has small or zero correlation.

A Signal Recognizer is applicable to signals which can be expressed as a series of products; that is,

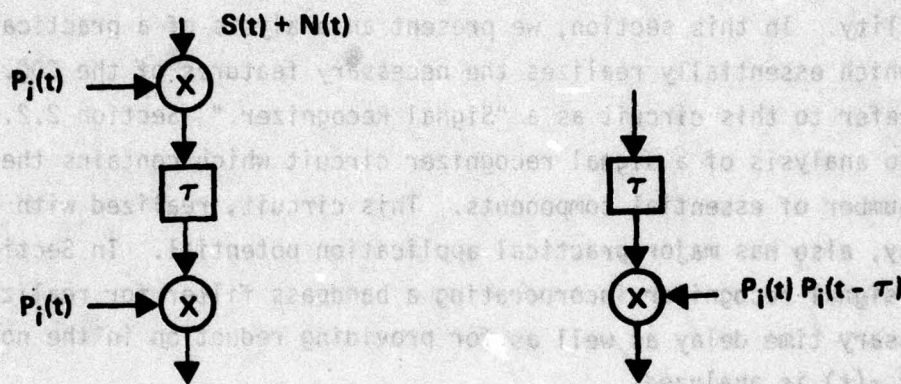
$$S(t) = P_1(t)P_2(t)\dots P_n(t) .$$

2-22

This circuit can be redrawn for a more practical implementation using the LDD designed for an AM desired signal modulation. In that example, it is assumed that the carrier frequency is the a priori known discriminant. Since it is carrier frequency we wish to track in this application, an amplitude modulation LDD can be ideally applied.

Theory

In the previous section, we discussed the properties of an ideal correlation discriminator operator without regard to physical realizability. In this section, we present a practical circuit which essentially realizes the necessary features. We will refer to this circuit as a "signal recognizer". Section 2.2.1 is devoted to analysis of a signal recognizer circuit which contains the minimum number of essential components. This circuit, used with a time delay, also has major practical application potential. In Section 2.2.2, a signal recognizer circuit is presented which is designed to provide the necessary time delay as well as providing reduction in the noise component $n(t)$ as analyzed.



Signal Recognizer Theory

Two practical signal recognizer circuits are illustrated in Figure 2.2-1. As shown in the figure, the signal recognizer consists of a multiplier to the a priori known function, delay in time by the amount τ , and multiplication again by the identical a priori known function. Equivalently, one can delay the signal of interest by an amount τ and then multiply by the combined product of a priori known function times delayed a priori known function. In a previous or later discussion, the time delay τ will be chosen so that the a priori known function times the delayed a priori known function has small or zero correlation. A signal recognizer is applicable to signals which can be expressed as a series of products, that is,

In order that a signal recognizer function as a CDO, the following conditions must be satisfied

$$0 = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} P_i(t) \dot{N}(t-\tau) dt \quad 2-23$$

$$0 = \lim_{T \rightarrow \infty} \int_t^{t+T} P_i(t) P_i(t-\tau) N(t) N(t-\tau) dt \quad 2-24$$

where $P_i(t)$ is the selected a priori discriminant term and $N(t)$ is any coherent interference plus any incoherent noise present in the system. Thus, the term $N(t)$ is composed of everything that is not desired signal.

Equation (2-23) requires that the a priori known function be unique to the signal and is the usual assumption made in spread spectrum communications. Equation (2-24) is somewhat more restrictive in that the product of the a priori function times a delayed version of the same function must also not be correlated with noise times delayed noise. In many applications this is a rather esoteric requirement. In the intended application, however, such correlations could inadvertently exist. An example will be given in Section 2.3.2, an AM signal recognizer example.

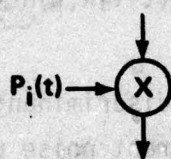
Finally, if useful desired signal correlations are to be obtained we must also have that

$$P_j(t-\tau) \approx P_j(t) . \quad 2-25$$

Here, the subscript j refers to any of the products comprising the signal which are not the a priori known function. In essence, this requirement means that the delayed partial signal can not be too different from the undelayed version of the signal; therefore if $P_j(t)$ changes faster than $P_i(t)$, $P_j(t)$ must be periodic. In practice, this requirement of (2-25) also sets limits on the maximum value that can be obtained by the signal recognizer time delay τ .

Figure 2.2-2 follows signal plus noise schematically through the signal recognizer. It is assumed that a waveform consisting of

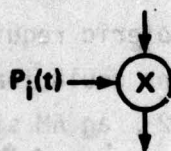
$$S(t) + n(t) = P_i(t)q(t) + n(t)$$



$$P_i^2(t)q(t) + n(t)P_i(t)$$



$$P_i^2(t - \tau)q(t - \tau) + n(t - \tau)P_i(t - \tau)$$



$$r(t) = P_i(t)q(t - \tau)P_i^2(t - \tau) + n(t - \tau)P_i(t)P_i(t - \tau)$$

Figure 2.2-2. Mathematical Representation of Waveforms Present in a Signal Recognizer

signal plus noise is applied to the signal recognizer. Here, we have expressed the signal as the product of the a priori known function $P_i(t)$ with the remainder of the signal which is referred to here as $q(t)$. If one follows through the time delay and second multiplication function, one will find that the reference signal is as given at the bottom of the figure. We have

$$r(t) = \left[P_i(t)q(t-\tau)P_i^2(t-\tau) \right] + \left[n(t-\tau)P_i(t)P_i(t-\tau) \right] . \quad 2-26$$

Notice that the term proportional to the desired signal contains the i^{th} product squared; even though this function is delayed it matters little because this is a power term. On the other hand, the term proportional to noise contains the known a priori function times itself delayed; by design we will ensure that this product has small or zero expected value.

It is now desirable to calculate the correlation of the incident composite waveform (the vector $x(t)$) with the reference function obtained from the signal recognizer. We start by calculating the correlation of incident signal $S(t)$ with $r(t)$. Using (2-26) we get

$$\begin{aligned} \frac{1}{T} \int_t^{t+T} S(t)r(t)dt &= \frac{1}{T} \int_t^{t+T} P_i^2(t)P_i^2(t-\tau)q(t)q(t-\tau)dt \\ &+ \frac{1}{T} \int_t^{t+T} P_i^2(t)P_i(t-\tau)n(t-\tau)dt . \end{aligned} \quad 2-27$$

We note that the i^{th} product squared is a positive nonzero quantity plus an alternating term. Defining the expected value of P_i^2 as C_i^2 and using the fact that P and q are assumed independent allows (2-27) to be simplified, we get

$$\frac{1}{T} \int_t^{t+T} S(t)r(t)dt = C_1^4 \int_t^{t+T} q(t)q(t-\tau)dt + C_1^2 \int_t^{t+T} P_1(t-\tau)n(t-\tau)dt \quad 2-28$$

According to our basic assumption, the i^{th} product is a unique signal discriminant. Therefore, using Equation (2-23), the second integral is zero. We get

$$\frac{1}{T} \int_t^{t+T} S(t)r(t)dt \approx C_1^4 C_q^2 \quad 2-29$$

where C_q^2 is defined as the expected value of $q(t)q(t-\tau)$. We have

$$C_q^2 = \frac{1}{T} \int_t^{t+T} q(t)q(t-\tau)dt \neq 0. \quad 2-30$$

Evidently then, it is necessary to choose the time delay τ so that the signal term $q(t)$ when delayed by an amount τ is still approximately equal to that function without a delay. If the desired signal were an amplitude modulated waveform and the a priori known function were the carrier term, then the term $q(t)$ would represent the modulation function. If the time delay τ is, let us say, a quarter of an RF cycle, then it is evident that the modulation is trivially changed in this time. Thus

$$q(t) \approx q(t-\tau). \quad 2-31$$

Having detailed the conditions necessary for the signal vector to correlate with the array reference function, we now determine the conditions necessary for ensuring that the noise waveform vector does not correlate with the array reference function.

Correlation of noise with the reference function is calculated as follows:

$$\frac{1}{T} \int_t^{t+T} n(t)r(t)dt = \frac{1}{T} \int_t^{t+T} n(t)P_i^2(t-\tau)P_i(t)q(t-\tau)dt$$

$$+ \frac{1}{T} \int_t^{t+T} n(t)n(t-\tau)P_i(t)P_i(t-\tau)dt .$$

2-32

Again, the first integral is zero according to the assumption of a unique signal discriminant Equation (2-23). The second integral, expressed earlier as (2-24), is fundamental to the theory of the correlation discriminant operator. Specifically, it establishes the condition that CDO output noise will not be correlated with any other noise in the system. Recall we stated earlier that if a signal recognizer is to function as a CDO, this integral must be forced to zero. Under ordinary circumstances, independence of $n(t)$ and $P(t)$ is sufficient to ensure zero correlation.

More importantly, the system designer can control the term $P_i(t)P_i(t-\tau)$ so as to ensure that the term has zero expected value. Then unless in some way alternating components of this product can in some way correlate with alternating terms in $n(t)n(t-\tau)$ the entire integral will be zero.

In general, the product of noise with delayed noise will not be zero. We have

$$n(t)n(t-\tau) = C_n^2 + h(t) \quad 2-33$$

where the term $C_n^2 \neq 0$. The term $h(t)$ is an alternating quantity which has zero average value. Similarly, the a priori product may be expressed as

$$P_i(t)P_i(t-\tau) = C_p^2 + f(t) . \quad 2-34$$

If the a priori discriminant waveshape and the time delay τ are appropriately chosen then we can ensure that

$$C_p^2 = 0$$

2-35

Alternatively, if $p(t)$ is a pseudo random sequence generated from an m bit register, then $\tau=1$ chip period will result in C_p^2 having the value $(2^m-1)^{-1}$; therefore, if the sequence is sufficiently long C_p^2 is approximately zero.

In most practical applications, it is sufficient only to require that C_p^2 be small or zero in order to ensure that the signal recognizer noise term $n(t)$ is uncorrelated as desired. Nevertheless, one must be aware that a nonzero expected value for this correlation can be obtained even though $C_p^2=0$ if the functions $f(t)$ and $h(t)$ are correlated. In the discussion of the amplitude modulation signal recognizer, we show how such correlation can exist. It is then pointed out that such correlation can usually be eliminated by appropriate filtering either in the signal recognizer or following it.

2.2.2 Signal Recognition Using a Bandpass Filter

In those cases where a multiplication by the chosen a priori discriminant function results in a spectrum despread (for example direct spread pseudo random modulation) the essential time delay of the signal recognizer can be realized with a bandpass filter. This filter may provide an additional advantage by reducing the amplitude of the uncorrelated noise term $n(t)$. However, in Chapter 3.0, Algorithms, we note that in some cases excessive delay in the signal recognizer can be detrimental to fast adaptation. Thus, a compromise between reduction of the noise term and signal recognizer delay may be necessary. In those applications requiring very short delays, an equivalent filter might only obtain a few dB reduction in the noise term and thus its additional complexity over that of the time delay may not be warranted.

In this section we will derive equations for a "Bandpass filter" signal recognizer then make performance comparisons with the "time delay" signal recognizer analyzed in the previous section. We refer to Figure 2.2-3 which illustrates this signal recognizer. For simplicity of analysis and the superior insight allowed, we will consider a direct spread signal throughout this derivation. We have

$$S(t) = AC(t)d(t)\cos(\omega_0 t + \phi) \quad 2-36$$

and noise $n(t)$. The output of the recognizer, which we shall again call $r(t)$ can be written as

$$\begin{aligned} r(t) &= C(t) \cdot \left\{ [S(t)C(t) + n(t)C(t)] * h(t) \right\} \\ &= C(t) \left\{ [S(t)C(t)] * h(t) \right\} + C(t) \left\{ [n(t)C(t)] * h(t) \right\} \\ &= r_s(t) + r_n(t) \end{aligned} \quad 2-37$$

where r_s and r_n are the signal and noise components respectively, and the symbol $*$ denotes convolution. Considering first the signal component and substituting Equation (2-36) into (2-37) we have

$$r_s(t) = C(t) \int_{-\infty}^{\infty} AC^2(\sigma)d(\sigma)\cos(\omega_0 \sigma + \phi)h(t-\sigma)d\sigma \quad 2-38$$

Taking Fourier transforms of the above expression results in

$$R_s(f) = C(f) * \left\{ A \left[\frac{D(f+f_0)}{2} e^{-j\phi} + \frac{D(f-f_0)}{2} e^{j\phi} \right] H(f) \right\} \quad 2-39$$

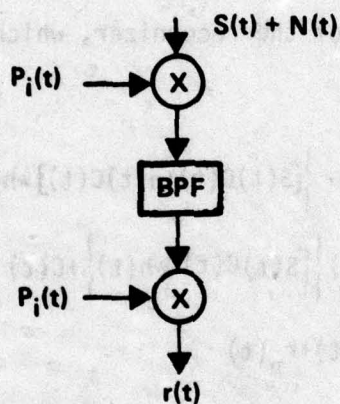


Figure 2.2-3. A Signal Recognizer Using the Time Delay of a Bandpass Filter

Figure 2.2-4 illustrates the frequency response of H, the amplitude of which is assumed to be constant over the bandwidth of D(f) but with a phase of $\theta + 2\pi(f - f_0)\tau$ in the neighborhood of f_0 . Hence equation (2-39) can be written as

2-40

$$R_s(f) = C(f) * \frac{AH(f_0)}{2} \left\{ D(f+f_0)e^{j[\theta-\phi-2\pi(f+f_0)\tau]} + D(f-f_0)e^{-j[\theta-\phi+2\pi(f-f_0)\tau]} \right\}.$$

Taking the inverse transform

$$r_s(t) = AH(f_0)C(t) d(t-\tau)\cos(\omega_0 t - \theta + \phi). \quad 2-41$$

We note that the linear phase characteristic of the filter H in the neighborhood of f_0 provides a time delay to the data d and a phase offset of θ provides a phase shift in the carrier.

If the filter H were a true time delay of delay τ then the only difference in the response relative to the above result is that θ would be replaced by $\omega_0 \tau$.

By inspection, (2-41) is essentially identical to the time delay result and signal-signal, signal-noise correlations found earlier (Equations 2-29 and 2-28) apply.

Focusing now on the noise response of the signal recognizer it is evident that the output is

$$r_n(t) = C(t)\{n(t)C(t)*h(t)\}. \quad 2-42$$

We now demonstrate that if there exists a linear phase component of the transform of $h(t)$ greater than one chip time then the random sequence $C(t)$ is uncorrelated from $C(t)n(t)*h(t)$, where $n(t)$ is arbitrary. If

Figure 2.2-4 illustrates the frequency response of H , the amplitude of which is assumed to be constant over the bandwidth of $B(f)$ but with a phase of $\theta(f)$ in the neighborhood of f_0 . Hence equation (2-19) can be written as

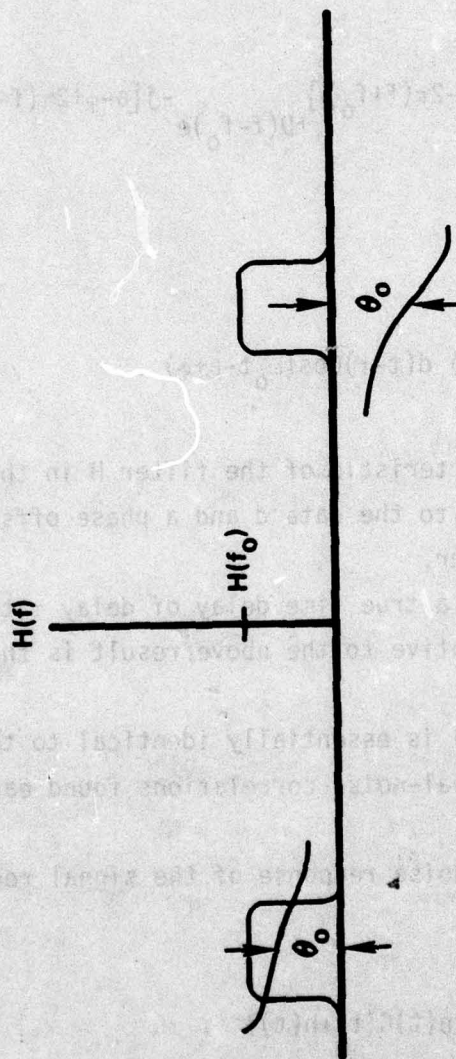


Figure 2.2-4. Assumed Bandpass Filter Amplitude and Phase Characteristics

we rewrite the expression in braces in terms of its Fourier transform the above expression can be written as

$$r_n(t) = C(t) \cdot F^{-1}\{F[C(t)n(t)] H(f)\} \quad 2-43$$

where $H(f)$ is the transform of $h(t)$. $H(f)$ can be decomposed into its amplitude and phase terms

$$H(f) = A(f)e^{-j\phi(f)} \quad 2-44$$

which can be written in the form

$$H(f) = e^{-j2\pi f\tau} A(f)e^{-j[\phi(f)-2\pi f\tau]} \quad 2-45$$

where τ is an arbitrary time such that

$$A(f)e^{-j[\phi(f)-2\pi f\tau]} \quad 2-46$$

is a realizable frequency response.

Substituting into (2-43) we have

$$r_n = C(t) \cdot F^{-1}\{F[C(t)n(t)]e^{-j2\pi f\tau} A(f)e^{-j[\phi(f)-2\pi f\tau]}\} \quad 2-47$$

Recognizing that

$$F\{C(t)n(t)\}e^{-j2\pi f\tau} = F\{C(t-\tau)n(t-\tau)\} \quad 2-48$$

we can write the above expression as

$$r_n(t) = C(t) \int_0^\infty C(t-\sigma-\tau) n(t-\sigma-\tau) h(\sigma) d\sigma \quad 2-49$$

where H is the inverse transform of $A(f)e^{-j[\phi(f)-2\pi f\tau]}$. Taking the average value over C we obtain

$$\overline{r_n(t)} = \int_0^\infty R_C(\sigma+\tau) n(t-\sigma-\tau) H(\sigma) d\sigma \quad 2-50$$

If τ is larger than a chip time it is evident that

$$R_C(\sigma+\tau) \approx 0 \quad \text{for all } \sigma \quad 2-51$$

hence

$$\overline{r_n(t)} \approx 0 \quad 2-52$$

If $A(f)e^{-j[\phi(f)-2\pi f\tau]}$ does not represent a realizable response then the lower limit of Equation (2-49) changes. We get

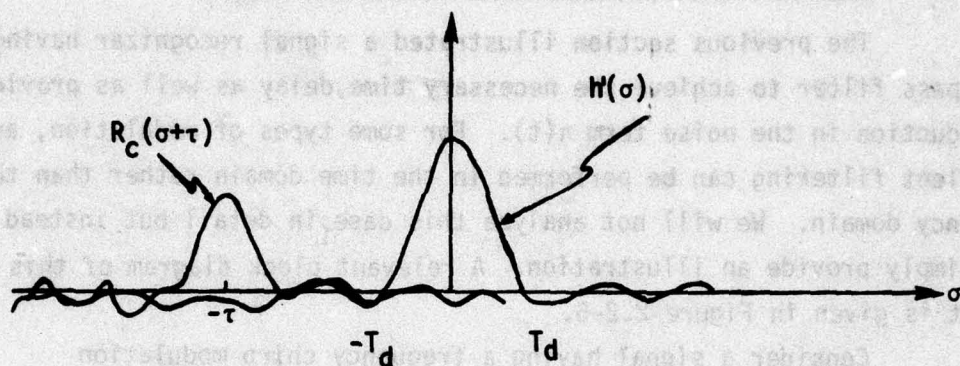
$$r_n(t) = C(t) \int_{-\infty}^\infty C(t-\sigma-\tau) n(t-\sigma-\tau) H(\sigma) d\sigma$$

Again taking the average over C yields

2-53

$$\overline{r_n(t)} = \int_{-\infty}^\infty R_C(\sigma+\tau) n(t-\sigma-\tau) h(\sigma) d\sigma \quad 2-54$$

If the integral is approximately zero then clearly $\overline{r_n(t)}$ will be nearly zero. To aid in the evaluation of the above expression consider the following figure which illustrates the two functions R_c and h



If τ is much larger than T_d (data bit time) then $\overline{r_n(t)}$ will be fairly small.

It is evident that if the filter h was a pure time delay of delay τ where τ is greater than a chip time, then H in Equation (2-54) would be replaced by unity.

Consider now the correlation of the noise component of the recognizer output with the input signal and noise. Focusing first on

$$r_n(t) \cdot S(t) = AC^2(t)d(t)\cos(\omega_0 t + \phi) \int_{-\infty}^{\infty} C(t-\sigma-\tau)n(t-\sigma-\tau)H(\sigma)d\sigma \quad 2-55$$

it is noted that if $C(t)$ or $n(t)$ have zero mean and are independent then the average of $r_n(t)S(t)$ is zero. Considering now the correlation of the noise component of the signal recognizer and the input noise we have

$$\overline{r_n(t) \cdot n(t)} = \int_{-\infty}^{\infty} R_c(\sigma+\tau)R_n(\sigma+\tau)H(\sigma)d\sigma. \quad 2-56$$

Again if τ is much larger than T_d or if the product of $h(\sigma)$ with $R_n(\sigma+\tau)$ is small, then the average value of $r_n(t)n(t)$ is near zero. These same conclusions hold if H is a pure time delay.

2.2.3 Signal Recognition Using a Time Domain Filter

The previous section illustrated a signal recognizer having a bandpass filter to achieve the necessary time delay as well as provide for reduction in the noise term $n(t)$. For some types of modulation, an equivalent filtering can be performed in the time domain rather than the frequency domain. We will not analyze this case in detail but instead will simply provide an illustration. A relevant block diagram of this circuit is given in Figure 2.2-5.

Consider a signal having a frequency chirp modulation (quadratic phase). We have

$$S(t) = A_m(t) \cos(\omega_0 t + \frac{at^2}{2}) \quad 2-57$$

A dispersive delay line matched filter for this signal can be constructed in several different ways. The output of such a matched filter is a $\frac{\sin(x)}{x}$ function, the pulse width of which is roughly inversely proportional to the duration of the chirp T . The pulse appears at the output of the matched filter about T seconds after the signal is applied. If the matched filter output is gated "on" whenever an output is expected and gated "off" otherwise, then noise power will be reduced by the ratio of "on" to "off" time while the signal power is virtually unaffected.

The signal recognizer is completed with the introduction of an inverse matched filter following the gate which causes the burst of signal energy to be rechirped. Observe that a multiplication by $P(t)P(t-\tau)$ is required. Details of this function are found in Section 2.3.3.

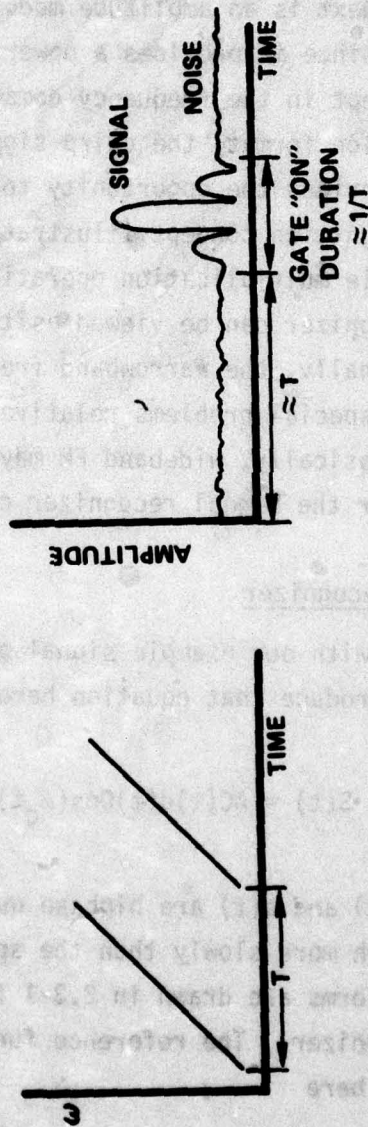
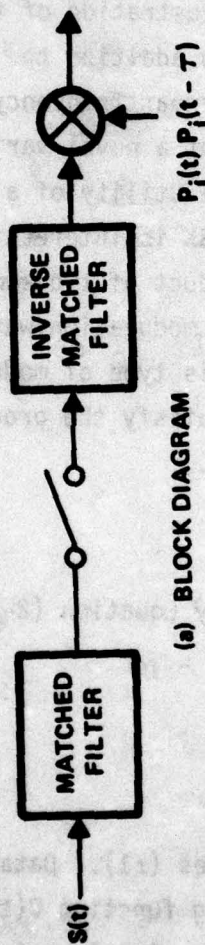


Figure 2.2-5. A Signal Recognizer Using Time Domain Filtering

2.3 Signal Recognizer Examples

In this section, we give specific examples of the signal recognizer. A PSK spread spectrum signal, because of its well defined correlation properties and precise phase shift characteristics, is a good initial example. Next is an amplitude modulation signal recognizer, also of great interest since it provides a powerful illustration of the signal recognition concept in the frequency domain. In addition to having a unique modulation format, the chirp signal (linear frequency shift vs. time) also provides the opportunity to present a novel variation of the signal recognition concept illustrating the utility of a single rather than double multiplication operation. FSK is interesting in that its signal recognizer can be viewed as the product of two simpler signal recognizers. Finally, the narrowband frequency modulation waveform is considered and special problems relative to this type of modulation are identified. Basically, wideband FM may not satisfy the product formulation required for the signal recognizer concept.

2.3.1 PSK Signal Recognizer

We continue with our example signal given by Equation (2-1). For convenience, we reproduce that equation here.

$$S(t) = AC(t)d(t)\cos(\omega_c t) . \quad 2-1$$

In the signal, both $C(t)$ and $d(t)$ are biphase quantities (± 1). Data information changes much more slowly than the spreading function $C(t)$ changes. Example waveforms are drawn in 2.3-1 for this signal as is a suitable signal recognizer. The reference function $r(t)$ is given by (2-26) which we repeat here

$$r(t) = P_i(t)q(t-\tau)P_i^2(t-\tau) + P_i(t)P_i(t-\tau)n(t-\tau) . \quad 2-26$$

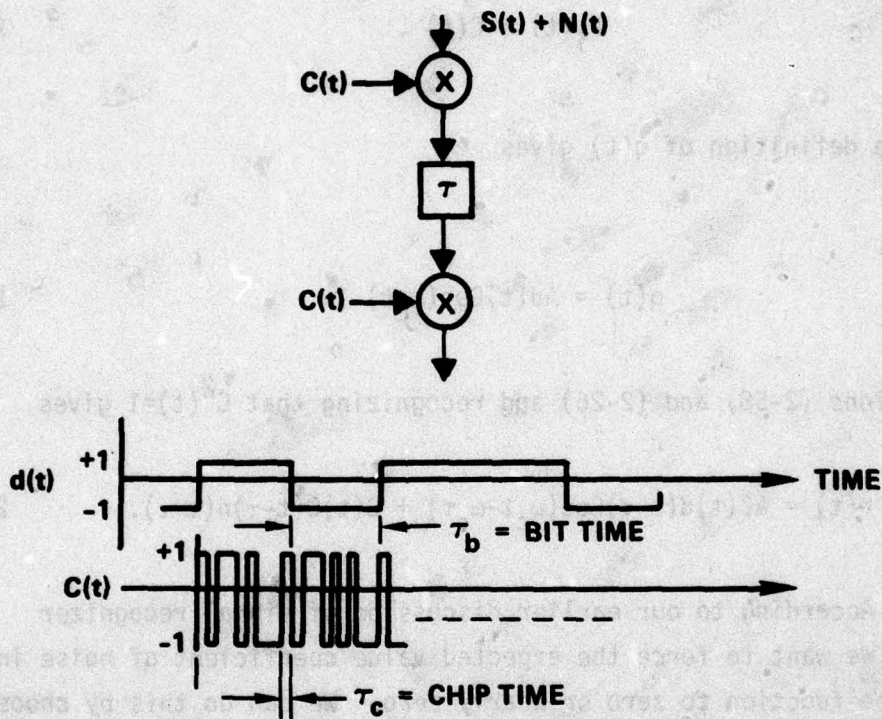


Figure 2.3-1. Biphase PSK Signal Recognizer and Relevant Waveforms

Recall that the term $q(t)$ is the product of all terms comprising the desired signal except for the a priori discriminant function. Let us select the pseudo random sequence $C(t)$ as the a priori discriminant. We have

$$P_1(t) = C(t) . \quad 2-58$$

Applying the definition of $q(t)$ gives

$$q(t) = Ad(t)\cos(\omega_c t) . \quad 2-59$$

Using Equations (2-58) and (2-26) and recognizing that $C^2(t)=1$ gives

$$r(t) = AC(t)d(t-\tau)\cos(\omega_c t - \omega_c \tau) + C(t)C(t-\tau)n(t-\tau) . \quad 2-60$$

According to our earlier discussion of signal recognizer principles, we want to force the expected value coefficient of noise in the reference function to zero or nearly zero. We can do this by choosing τ as follows

$$\tau \geq \tau_c \quad 2-61$$

where the term τ_c is identified in Figure 2.3-1 as the period of a pseudo random chip. In this case, the expected value coefficient of the noise term is $(2^m-1)^{-1}$ where m is the number of bits in the pseudo random generator. Thus, relatively short code sequences can ensure negligible correlation terms.

In order to preserve reasonable desired signal correlations, we must also satisfy Equation (2-30) which states that $q(t)q(t-\tau)$ must

correlate. For data having more or less equal probability of sign, this means that the signal recognizer time delay τ cannot exceed the duration of one data bit, τ_b ; thus, one must also require that

$$\tau \leq \tau_b.$$

2-62

Finally, some algorithms, particularly the PSF discussed in Chapter 3.0, require that phase be preserved in the signal recognition operation. This constraint additionally requires

$$\omega_c \tau = k(2\pi) \quad k = \text{an integer}$$

2-63

Before leaving this example, several comments are warranted. First, if it is desirable to maximize desired signal correlations, then time delay τ should be just slightly greater than 1 chip time. Alternatively, if suppression of the uncorrelated noise term $n(t)$ is of paramount importance, then realization of the time delay with a bandpass filter is recommended. In this case, selection of τ as approximately 1/2 bit time sacrifices only 3 dB of desired signal correlation while achieving only 3 dB less than the maximum possible rejection of noise. We note, however, that ordinarily bit times are relatively long and maintaining the phase stability of such a filter may be difficult. Furthermore, some very fast adapting algorithms may not suitably employ such long delay signal recognizers.

We wish to make another important point regarding suppression of the undesirable noise term correlations. Even if the noise term $n(t)$ produces non-zero correlations, such correlations may not be greatly detrimental to an adaptive processor. As long as the reference signal provided to the error formation difference port has correlated noise terms with amplitude less than that required to change the sign of the net noise correlations, then the adaptive array will suppress these terms regardless.

Mathematically, the result obtained is equivalent to a reduction in the power of those correlated noise components. Correlation in $n(t)$ as large as 1/2 as referenced to the array error formation difference port would effectively reduce a jammer's power by 3 dB. Ordinarily, this would only modestly reduce null depth performance of the adaptive processor.

Therefore, even though the CDO and its practical realization, the signal recognizer, can theoretically produce arbitrarily small or zero correlation noise terms, considerable tolerance in the practical realization is allowed.

Instead of having chosen the term $C(t)$ as the a priori known discriminant for this example, we could also have chosen the carrier term $\cos(\omega_c t)$. Had we made this selection the signal recognizer would be the same as that discussed next, the Amplitude Modulation Signal Recognizer.

2.3.2 AM Signal Recognizer Example

We begin with the equation of an amplitude modulated signal:

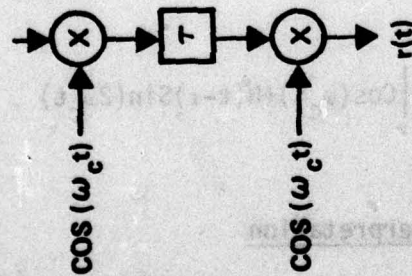
$$S(t) = A[1+m(t)]\cos\omega_c t \quad 2-64$$

Here we take the term $\cos\omega_c t$ to be the a priori known signal discriminant. The signal recognizer is then illustrated in Figure 2.3-2. In order to complete the design of this recognizer, we must now specify the magnitude of the time delay τ .

As in the previous example we require the expected value coefficient of noise in $r(t)$ to be zero. That is

$$\frac{1}{T} \int_t^{t+T} P_i(t)P_i(t-\tau)dt = 0 \quad 2-65$$

SIGNAL RECOGNIZER



EQUIVALENT CIRCUIT

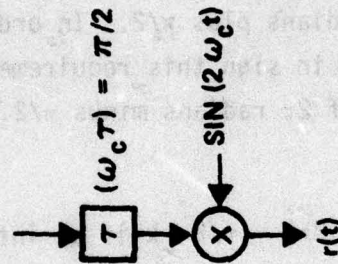


Figure 2.3-2. A Signal Recognizer for an AM Desired Signal

For this case, the specific equation is

$$\frac{1}{T} \int \cos(\omega_c t) \cos \omega_c (t-\tau) dt = \left(\frac{\cos \omega_c \tau}{2} \right). \quad 2-66$$

In order for the term $\cos \omega_c \tau$ to be zero, we must require that the argument be an integral number of π radians plus $\pi/2$. In order that the "recognized" signal not be changed in sign this requirement is further restricted to an integral number of 2π radians minus $\pi/2$. We get

$$\omega_c \tau = 2\pi k - \pi/2 \quad k \text{ is an integer.} \quad 2-67$$

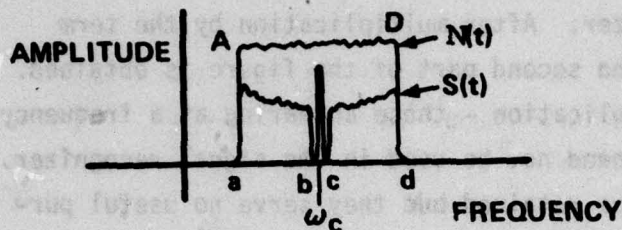
If we examine the signal recognizer in Figure 2.3-2(a) for the case where $\omega_c \tau = 3\pi/2$ then it is evident that we can redraw the circuit as illustrated in Figure 2.3-2(b). Here we have used the fact that $\cos(x)\sin(x)$ is equal to $\frac{1}{2}\sin(2x)$.

The reference function $r(t)$ using (2-26), (2-64) and (2-67) gives

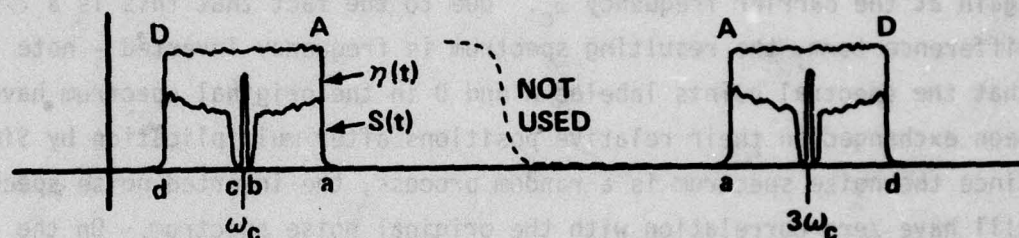
$$r(t) = \frac{A}{2} [1+M(t-\tau)] \cos(\omega_c t) + N(t-\tau) \sin(2\omega_c t). \quad 2-68$$

2.3.2.1 Frequency Domain Interpretation

A frequency domain interpretation of Equation (2-68) is given in Figure 2.3-3. Such an interpretation is very useful in that Equation (2-68) does not yield intuitive understanding of the signal recognizer operation.



AFTER MULTIPLICATION BY $\sin(2\omega_c t)$



- NOISE SPECTRUM IS FREQUENCY INVERTED, UPPER AND LOWER SIDEBANDS HAVE ZERO CORRELATION
- SIGNAL SPECTRUM IS ALSO INVERTED, BUT SINCE IT IS SYMMETRICAL ABOUT ω_c , IT APPEARS UNCHANGED

MULTIPLE SOURCES (INTERFERENCE $\omega_{ci} \neq \omega_{cs}$)

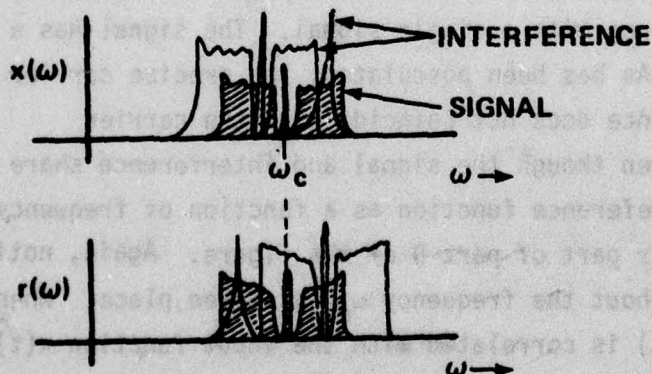


Figure 2.3-3. Relevant Spectra for an AM Signal Recognizer

At the top of the figure is spectrum amplitude plotted versus frequency for both uniform noise and the amplitude modulated signal. Several points on the spectrum have been labeled in order to illustrate the action of the signal recognizer. After multiplication by the term $\sin 2\omega_c t$, the spectrum shown on the second part of the figure is obtained. The sum frequencies of the multiplication - those appearing at a frequency 3 times the carrier frequency - need not be used in the signal recognizer. These terms could theoretically be retained but they serve no useful purpose. On the other hand, the difference frequency terms as noted appear again at the carrier frequency ω_c . Due to the fact that this is a difference term, the resulting spectrum is frequency inverted - note that the spectral points labeled A and D in the original spectrum have been exchanged in their relative positions after multiplication by $\sin 2\omega_c t$. Since the noise spectrum is a random process, the inverted noise spectrum will have zero correlation with the original noise spectrum. On the other hand, the signal spectrum is also inverted, but since it was initially symmetrical, it is unchanged by the inversion. Consequently, the signal in the reference function will be strongly correlated with the input signal.

The case of multiple interferences is illustrated in Figure 2.3-3(b). In the input spectrum which is labeled as $x(\omega)$, two interferences are shown along with a single signal. The signal has a carrier at frequency ω_c . As has been postulated, the precise carrier frequency of the interference does not coincide with the carrier frequency of the signal even though the signal and interference share the same bandwidth. The reference function as a function of frequency is illustrated in the lower part of part B of the figure. Again, notice that frequency inversion about the frequency ω_c has taken place. When the reference function $r(t)$ is correlated with the input function $x(t)$, it is evident that only the signal term will yield average value. Beat frequencies will be obtained for the interferences due to the fact that frequency inversion was performed about a frequency not equal to their carrier frequencies.

We show this mathematically by assuming an input function of the following form:

$$x(t) = S(t) + I(t) \quad 2-69$$

where we have

$$x(t) = A[1+m(t) \cos \omega_c t] + \sum_j A_j f_j(t) \cos \omega_j t. \quad 2-70$$

Note that the interferences have frequencies given by $\cos \omega_j t$. The signal recognizer provides a reference function $r(t)$ given by the following:

$$r(t) = A[1+m(t-\tau)] \cos \omega_c t + \sum_j A_j f_j(t-\tau) \cos[(2\omega_c - \omega_j)t + \theta_j] \quad 2-71$$

where $\theta_j = -\omega_j \tau$.

We now calculate the correlation of the reference function with the input signal vector. We obtain

$$\begin{aligned} \frac{1}{T} \int_t^{t+T} r(t) S(t) dt &= \frac{A^2}{2T} \int_t^{t+T} [1+m(t)] [1+m(t-\tau)] dt \\ &+ \sum_j \frac{AA_j}{2T} \int_t^{t+T} [1+m(t)] f_j(t-\tau) \cos[(\omega_c - \omega_j)t + \theta_j] dt \end{aligned} \quad 2-72$$

Examination of the second integral shows that terms involving the frequency $(\omega_c - \omega_j)$ are present. Obviously, these terms have a zero average value.

Note that the first integral above evaluates the power in the modulation of the AM transmission. Since the modulation is a very slowly varying function in comparison with the rate of variation of the carrier

frequency, the modulation times delayed modulation is essentially equal to the modulation squared. That is, we have

$$E\{[1+m(t)][1+m(t-\tau)]\} \approx \frac{A^2}{2} E\{[1+m(t)]^2\} \quad 2-73$$

Therefore, it is evident that the reference function is highly correlated with the input desired signal.

We now calculate the correlation of the reference function with the k^{th} interference present in the input function $x(t)$. We have

$$\begin{aligned} \frac{1}{T} \int_t^{t+T} r(t) I_k(t) dt &= \frac{AA_k}{2T} \int_t^{t+T} f_k(t) [1+m(t-\tau)] \cos[(\omega_c - \omega_j)t + \theta_j] dt \\ &+ \sum_j \frac{A_j A_k}{2T} \int_t^{t+T} f_k(t) f_j(t-\tau) \cos[(2\omega_c - \omega_k - \omega_j)t + \theta_k + \theta_j] dt \end{aligned} \quad 2-74$$

The first integral again has terms containing the frequency difference $(\omega_c - \omega_j)$, and therefore has no average value. The second integral involves frequency differences $(2\omega_c - \omega_k - \omega_j)$, which are obtained from the cross products of the several interferors. This integral is zero provided that

$$\omega_c \neq \left(\frac{\omega_k + \omega_j}{2} \right) \quad 2-75$$

or that

$$E\{f_k(t) f_j(t-\tau)\} = 0. \quad 2-76$$

Since $f(t)$ is unity for a CW interference, assumed decorrelation is dependent upon Equation (2-75).

The reader may recall that in Section 2.2, we listed conditions required of a signal recognizer in order that it be a correlation discriminant operator. The second condition, expressed by Equation (2-24) is reasserted above in Equation (2-74). Specifically, Equation (2-24) requires that the discriminant function times delayed discriminant function not contain terms also present in interference times delayed interference. As can be seen, CW jammer intermodulation terms may produce an average value correlation even though the discriminant function itself has no correlation with the interference.

Interestingly, the requirement of Equation (2-75) can be removed if the reference function is filtered so as to pass only frequencies near the carrier frequency ω_c . While such an expedient is probably not necessary, it is certainly easy to do and does eliminate the possibility of such correlation products being obtained.

2.3.2.2 Comparison with a Customary Approach

It is interesting to compare the signal recognizer approach with a direct filtering approach which has been used by some investigators. Schematically, the technique is shown in Figure 2.3-4. Signal and noise is input to a narrowband filter centered about the carrier frequency, and the output of this filter is the reference function $r(t)$. Relevant spectral diagrams are given in Part B of the figure. The input signal and noise is shown in the upper diagram followed by the filter frequency response function. The reference function obtained is pictured in the bottom diagram. Comparison of the reference function with the input function clearly shows that the noise terms passed by the filter will correlate with the noise terms in the input. Therefore, it is evident that

$$\frac{1}{T} \int n(t)r(t)dt \neq 0 .$$

2-77

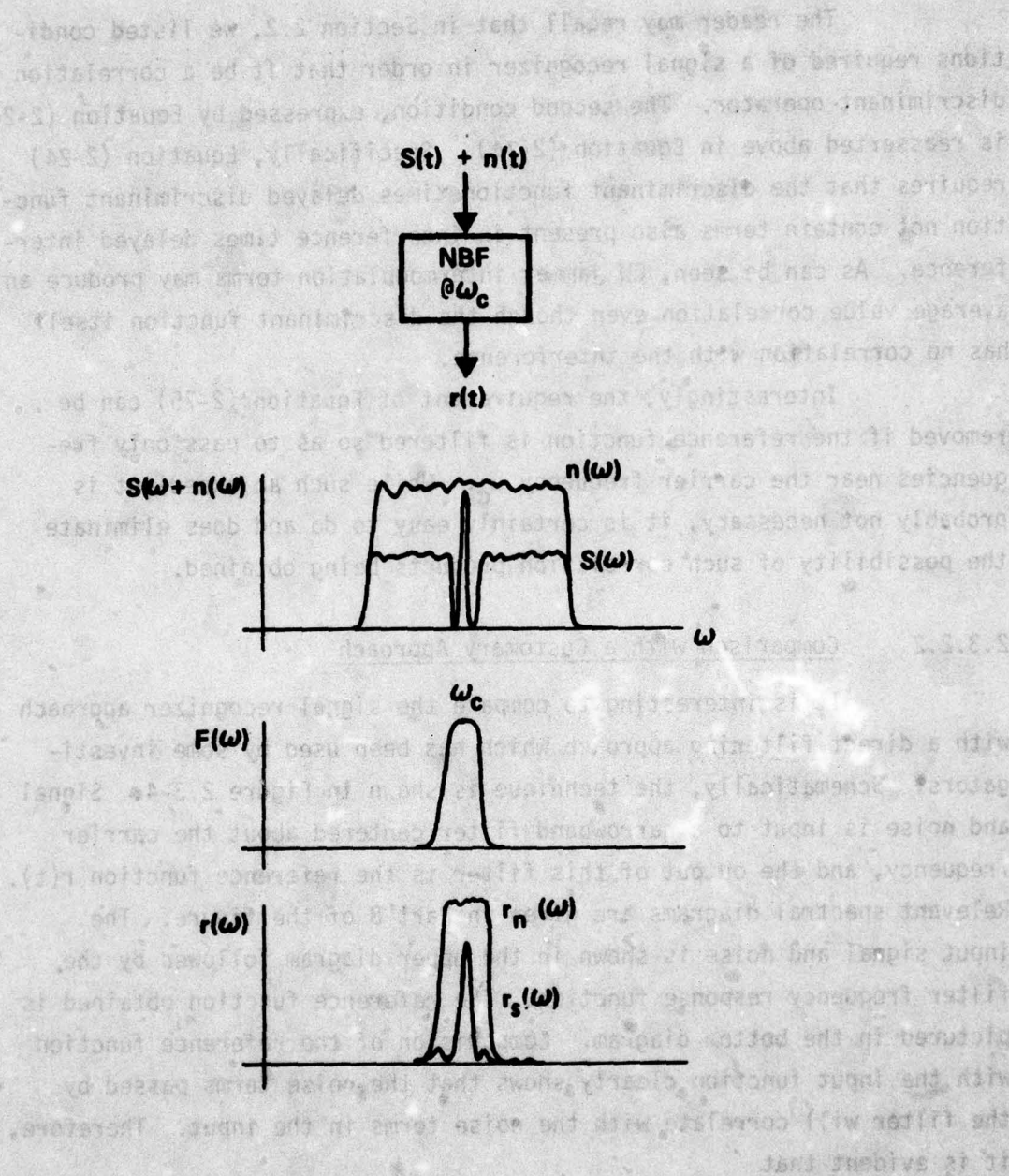


Figure 2.3-4. A Customary Approach for Obtaining an AM Desired Signal Reference Function

This means that a non-optimal array solution will be obtained. In fact, if the noise power is of the same bandwidth and density as the signal power, then no performance gain will be attained by the array.

2.3.3 A Chirp Signal Recognizer

A chirp signal is one in which the carrier frequency is changed linearly as a function of time. We begin with a mathematical representation of the signal. We have

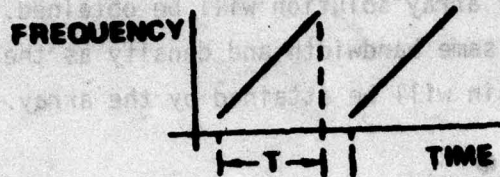
$$S(t) = d(t) \cos \left(\omega_c t + \left(\frac{\alpha t^2}{2} \right) \right) . \quad 2-78$$

This equation may be expanded to yield

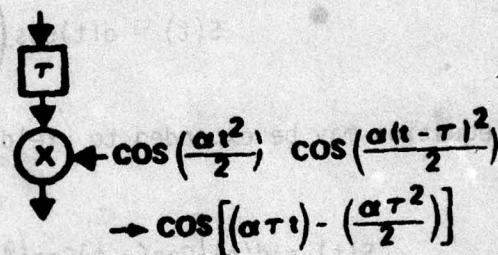
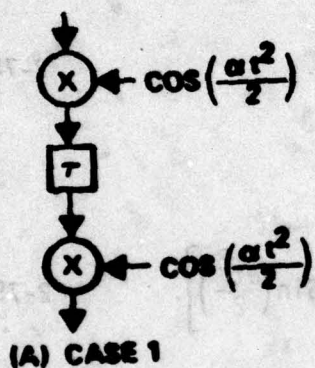
$$S(t) = d(t) \left[\cos(\omega_c t) \cos \left(\frac{\alpha t^2}{2} \right) - \sin(\omega_c t) \sin \left(\frac{\alpha t^2}{2} \right) \right] . \quad 2-79$$

In the above, the term responsible for the continuous change of frequency with time is $\frac{\alpha t^2}{2}$. Information is transmitted by the signal through the term $d(t)$. This can be a biphase modulation or conventional amplitude modulation. A spectral picture of the signal is given in Figure 2.3-5(a). The duration of the chirp is seen to be of time T . After this time has elapsed, the chirp function is restarted. Therefore, in the frequency domain such a signal has a sawtooth waveform as a function of time.

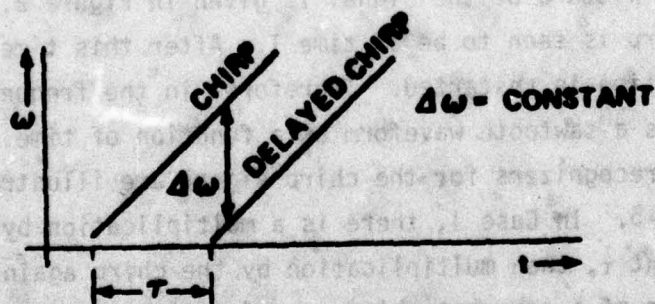
Two signal recognizers for the chirp signal are illustrated in Part B of Figure 2.3-5. In Case 1, there is a multiplication by the chirp, delay by an amount τ , then multiplication by the chirp again. Since precise generation of a coherent chirp as a function of time may be a fairly difficult matter, the realization shown in Case 2 may be preferred. Here, the multiplication function is the chirp times a delayed chirp. While at first this may appear to be a very difficult



(a) SIGNAL REPRESENTATION



(b) SIGNAL RECOGNIZER FUNCTIONAL DIAGRAMS



(c) INTERPRETATION OF CASE 2

Figure 2.3-5. Chirp Desired Signal Waveforms and Signal Recognizers

term to generate, expansion of the expression shows that the difference frequency is a constant frequency sinusoid. The difference frequency of this product is given mathematically by the expression:

$$\cos \left[\left(\alpha \tau t \right) - \left(\frac{\alpha \tau^2}{2} \right) \right] \quad 2-80$$

There is a phase term given by $\frac{\alpha \tau^2}{2}$ which is constant.

An interpretation of Case 2 is given in Part C of the figure where a chirp and a delayed chirp are plotted as a function of time. Notice that the time delay τ establishes the frequency difference between the two waveforms and that this difference is a constant.

Although it may not be evident, it is unnecessary for one to generate the sum frequency term of the chirp and delayed chirp product. It can be shown that disregard of this term leads merely to a loss of amplitude which can be made up with an amplifier.

Thus, the chirp recognizer is remarkably simple consisting of a time delay and a multiplication by a constant frequency sinusoid. We now calculate the amount of time delay which is required in order to produce noise decorrelation. As before, we require

$$\frac{1}{T} \int_0^T P(t)P(t-\tau)dt = 0 \quad 2-81$$

Thus, using expression (2-80) and (2-81) it is necessary that

$$(\alpha \tau T) = n(2\pi) \quad n \text{ is an integer} \quad 2-82$$

2.3.4 FSK Signal Recognizer Example

The equation for a frequency shift key modulated signal is

$$S(t) = A \left\{ \frac{(d(t)+1)}{2} \cos \omega_1 t + \frac{(d(t)-1)}{2} \cos \omega_2 t \right\} \quad 2-83$$

where data, $d(t)$, is biphase. Examination of Equation (2-83) shows that if the data function is equal to +1, then frequency ω_1 is transmitted. Similarly, if $d(t)=-1$, frequency ω_2 is transmitted. For convenience, we will write that frequencies ω_1 and ω_2 are formed by making a deviation from a carrier frequency ω_c as follows:

$$S(t) = A \left\{ \left(\frac{d(t)+1}{2} \right) \cos(\omega_c + \Delta\omega)t + \left(\frac{d(t)-1}{2} \right) \cos(\omega_c - \Delta\omega)t \right\} \quad 2-84$$

The signal recognizer circuitry for the FSK signal and several relevant spectra are shown in Figure 2.3-6. Expansion of Equation (2-84) will reveal that a good signal discriminant function consists of the carrier frequency and the frequency deviation quantity. As in previous signal recognizer designs, twice each of these frequencies is the appropriate term to be used. Furthermore, the time delay is chosen so as to ensure the decorrelation of noise when multiplied by these frequency separation parameters.

The signal spectrum is illustrated in Part A of Figure 2.3-6. The mid-part of the spectrum is at the carrier frequency ω_c . Peaks A and B correspond to transmission of data bits at frequencies ω_1 and ω_2 . Part B shows the spectrum at the multiplication by twice the carrier frequency. Notice that the spectral peaks B and A have been inverted in frequency.

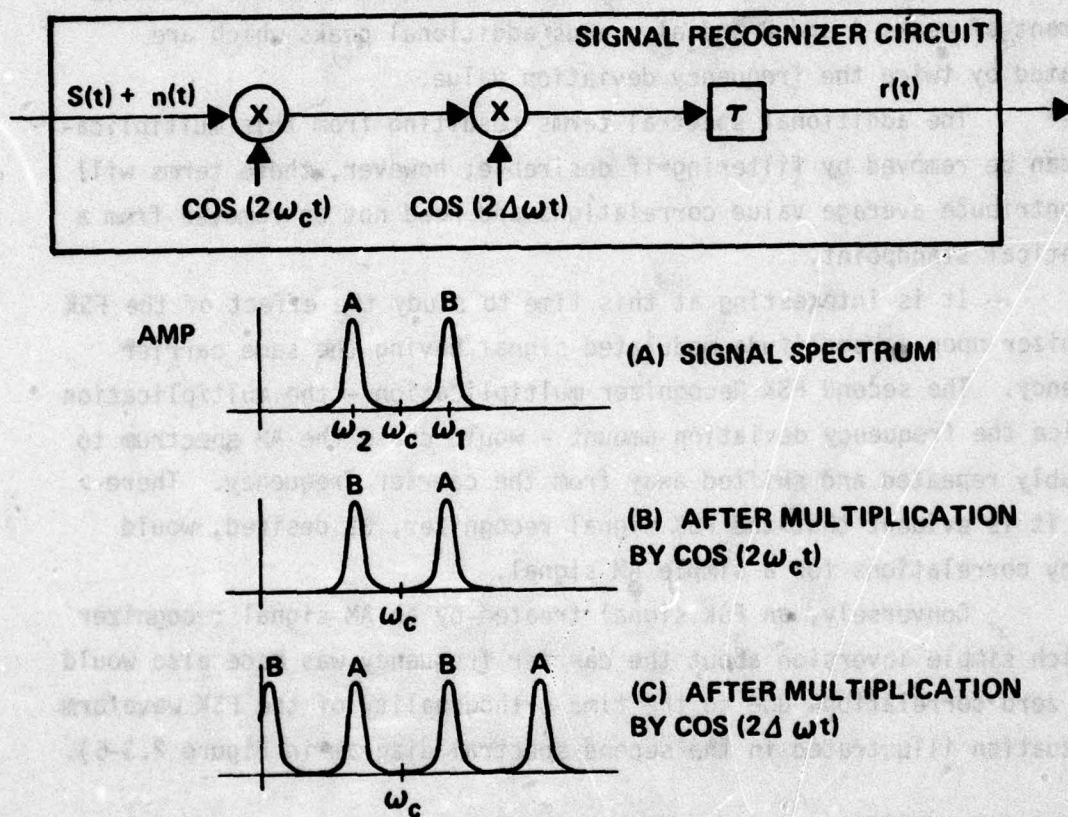


Figure 2.3-6. FSK Signal Recognizer and Relevant Spectra

At this point, a useful signal recognition function has not been obtained for the FSK signals. This is due to the fact that peaks A and B are orthogonal in time. That is to say, when a (+1) bit is being transmitted, a (-1) bit is not being transmitted; when A is present, B is absent. A correct orientation in the frequency domain is obtained with multiplication by $\cos(2\Delta\omega t)$. This multiplication restores the relative placement of peaks A and B and also adds additional peaks which are separated by twice the frequency deviation value.

The additional spectral terms resulting from this multiplication can be removed by filtering if desirable; however, these terms will not contribute average value correlations and need not be removed from a theoretical standpoint.

It is interesting at this time to study the effect of the FSK Recognizer upon an amplitude modulated signal having the same carrier frequency. The second FSK Recognizer multiplication - the multiplication by twice the frequency deviation amount - would cause the AM spectrum to be doubly repeated and shifted away from the carrier frequency. Therefore, it is evident that the FSK signal recognizer, as desired, would destroy correlations for a simple AM signal.

Conversely, an FSK signal treated by an AM signal recognizer in which simple inversion about the carrier frequency was made also would yield zero correlations due to the time orthogonality of the FSK waveform (a situation illustrated in the second spectral diagram in Figure 2.3-6).

2.3.5 FM Signal Recognizer Circuit

A general angle modulated signal can be described mathematically as follows:

$$S(t) = A \cos[\omega_c t + \theta(t)] \quad 2-85$$

where $\theta(t)$ contains the information being transmitted. Expansion of this function yields the form convenient for use in the signal recognizer.

$$S(t) = A\{\cos(\omega_c t)\cos\theta - \sin(\omega_c t)\sin\theta\} . \quad 2-86$$

In a narrowband FM modulation system, the quantity θ is usually small compared to 2π radians in magnitude and it has an average value of zero.

Inspection of Equation (2-86) reveals that we can use either $\cos(\omega_c t)$ or $-\sin(\omega_c t)$ as an a priori discriminant in the same manner as used in the AM signal recognizer. Regrettably this approach does not utilize all of the available signal power for recognition, and furthermore, this signal recognizer could also respond to an AM signal. In a wideband FM system, it is possible that the parameter $\theta(t)$ contains the a priori discriminant. A signal recognizer for this function would entail a sine or cosine transformation of the entire modulation information, which is unknown. Observe that biphase PSK or quadrature PSK represents special cases in that the $\sin[\theta(t)]$ or $\cos[\theta(t)]$ becomes ± 1 conveniently.

If a narrowband FM signal is being transmitted, such as might be found in the HF/VHF/UHF bands, the quantity θ is usually small compared to 2π radians in magnitude. Furthermore, let us assume that it has an average value of zero. Consequently, $\cos(\theta)$ is approximately unity and slowly varying. Alternatively, the term $\sin(\theta)$ is relatively rapidly varying and, in a narrowband FM system, is the term that is responsible for most of the bandwidth of the signal.

Using this information, it is possible to construct several different signal recognizers. In each case, however, compromise is necessary. Evidently one must either sacrifice desired signal correlations or allow undesired noise correlations to some extent. If the interference properties are known in advance, then a better FM signal recognizer design can be obtained. For example, selectively inverting the wide sidebands due to $\sin(\theta)$ while not inverting the narrow side-

bands proportional to $\cos(\theta)$ could provide substantial decorrelation to an AM interferer but this approach obviously would be ineffective against a jammer having a carrier frequency equal to that of the desired signal.

An important conclusion which can be drawn from this brief discussion is that only special cases of angle modulation such as PSK and QPSK are optimal for use with the signal recognizer approach.

3.0

ALGORITHMS

Numerous algorithms and their variations received consideration during this study. Two new algorithms were conceived and significant strides were taken in the area of perturbational algorithms, the result being the development of a technique for realizing any conventional direct correlation-type algorithm in a perturbational configuration.

Another significant result, a consequence of the development of the perturbational technique, is the discovery of an expected value equivalence between gradient algorithms and the conventional random search algorithm.

Since perturbational algorithms are generally slower than their direct correlation counterparts due to the fact that a single channel (the array output/receiver pathway) must carry all of the gradient information in multiplex form; it is important to make the algorithm as efficient in its adaptation as is feasible. In this regard, the correlation discriminant operator (CDO) is particularly useful since it can be used to enable unbiased estimation of desired signal terms in a minimum of time. Since a signal-to-noise ratio maximizing algorithm must base each of its steps upon achieved signal and noise power changes due to the previous step, signal power estimation time is a crucial parameter in the rate of adaptation. The modified perturbational PSF algorithm which we describe in Section 3.3 is significant in that it can achieve iteration rates and adaptation times principally limited by the bandpass of the receiver connected to the array output. This is orders of magnitude faster than conventional approaches.

Two algorithms, an Error Phase Tolerant (EPT) RF/IF direct correlation procedure which is stable for arbitrary phase shifts in the error channel and a novel self-orthogonalizing direct correlation procedure which promised to essentially equalize jammer and desired signal adaptation times, received brief consideration during this study but do not appear in this report. The reasons are as follows.

The need for the EPT procedure was obviated by development of the perturbational PSF algorithm. Since the error phase tolerant

procedure requires analog hardware correlators as well as frequency converters and amplifiers, hardware cost, size, weight and power are comparatively much greater than for the simple microprocessor-based perturbational PSF procedure. Sufficient analysis was not done to indicate whether the EPT algorithm might have substantially greater adaptation speed, and thus, be warranted in some critical applications.

The second algorithm, a self-orthogonalizing procedure, is apparently an inherently noisy technique. Since perturbational techniques also are considerably noisier than direct correlation procedures, it was quickly decided that a perturbational self-orthogonalizing algorithm was inappropriate in this application.

Consequently, this chapter is largely devoted to those algorithms which are directly applicable to the perturbational approach and which are general enough to address the system problem postulated in Chapter 4.0.

An algorithm which received attention early in our study which we refer to as "log power" is also briefly described due to the fact that it was the procedure implemented in the first microprocessor simulation (the 8080 based circuit). This algorithm is not discussed in detail, however, because it does not play a role in the selected system configuration.

We begin this presentation with a mathematical description of the PSF algorithm and its variations. The "direct correlation" differential equation descriptions are given first because they promote understanding of the procedures, whereas the perturbational descriptions are considerably more complex. Once the basic differential equations of these algorithms are understood, however, an understanding of the perturbational procedures is entirely straightforward.

Following this, we address the topic of perturbational correlation in Section 3.2. Basically, this section shows how the necessary information to generate the algorithm differential equations can be obtained through several different weight perturbation procedures. This presentation leads naturally to a consideration of random perturbations, and thus, to the random search algorithm in 3.3. Here, for the first

time, we reveal the expected value equivalence of the random search algorithm with a normalized gradient algorithm.

Also, in Section 3.3, we combine the basic differential equations with the results of the perturbational correlation techniques and show how the modified algorithms are achieved.

3.1 Basic Differential Equations

A general description of the PSF algorithm basic differential equations begins this section. After a brief treatment of an approximate solution to the nonlinear differential equation, we discuss important details of algorithm stability. It is noted that the algorithm is stable only at that weight vector which maximizes the output signal-to-noise ratio, but the proof is not totally complete in that if more than one exactly equivalent maxima is obtainable, the algorithm might transition between these equivalent solutions. We follow these stability considerations in Section 3.1.2 with a configurationally improved PSF algorithm capable of obtaining the same optimum results but having substantially reduced weight jitter, achieved by reduction of errors in the desired signal steering vector.

The generality of the PSF algorithm is then expanded in Section 3.1.3 to accommodate the case of multiple simultaneous desired signals.

3.1.1 Basic Positive Signal Feedback Algorithm

A new adaptive array algorithm using the least mean squares error criteria is described in this Section. This algorithm, referred to here as the Positive Signal Feedback Algorithm (PSF) improves upon the traditional LMS algorithm of Widrow et. al. Specifically, the PSF circuit provides for arbitrary independent adjustment of the desired signal adaptation time constant with respect to the adaptation time constants for jammers during the initial part of its transient; it provides for reversion to a suppression algorithm characterized by a well behaved antenna pattern in the absence of a desired signal; and it requires

weights having a dynamic range capability constrained only by relative pattern formation needs rather than including as well any additional incident signal power variations such as those due to signal fades or to varying range from a transmitter.

A block diagram of the PSF algorithm is given in Figure 3.1-1. Inspection of the diagram allows one to write the following differential equations:

$$\frac{dW}{dt} = -K_1 [R_X - \alpha R_S] W \quad 3-1$$

$$\frac{d\alpha}{dt} = K_2 W^T X [X^T W - \alpha S^T W] = K_2 W^T [R_X - \alpha R_S] W . \quad 3-2$$

In the above equations, $x(t)$ is the incident composite waveform vector, W is the weight vector, R_X is the incident waveform crosscorrelation matrix, R_S is the incident signal crosscorrelation matrix, K_1 and K_2 are scalar gain constants, and α is the signal feedback loop weight value.

The block labeled "Signal Recognizer" has been described in detail in Chapter 2.0.

Before the transient solution to the differential equations is discussed, let us examine the "steady state" solutions, i.e., $\frac{dW}{dt}$ and $\frac{d\alpha}{dt}$ both equal to zero. The term α is obtained from (3-2) and substituted into (3-1) yielding

$$R_X W = \left(\frac{W^T R_X W}{W^T R_S W} \right) R_S W . \quad 3-3$$

It can be shown that (3-3) is also obtained if one maximizes the broadband signal-to-noise ratio obtainable from the weighted combination of "n" weights. (See Section 3.3.2.) These "n" weight vector solutions can be calculated using eigenvalue analysis.

Comparatively, the broadband LMS algorithm having an a priori signal estimate $S_0(t)$ produces an array output most nearly

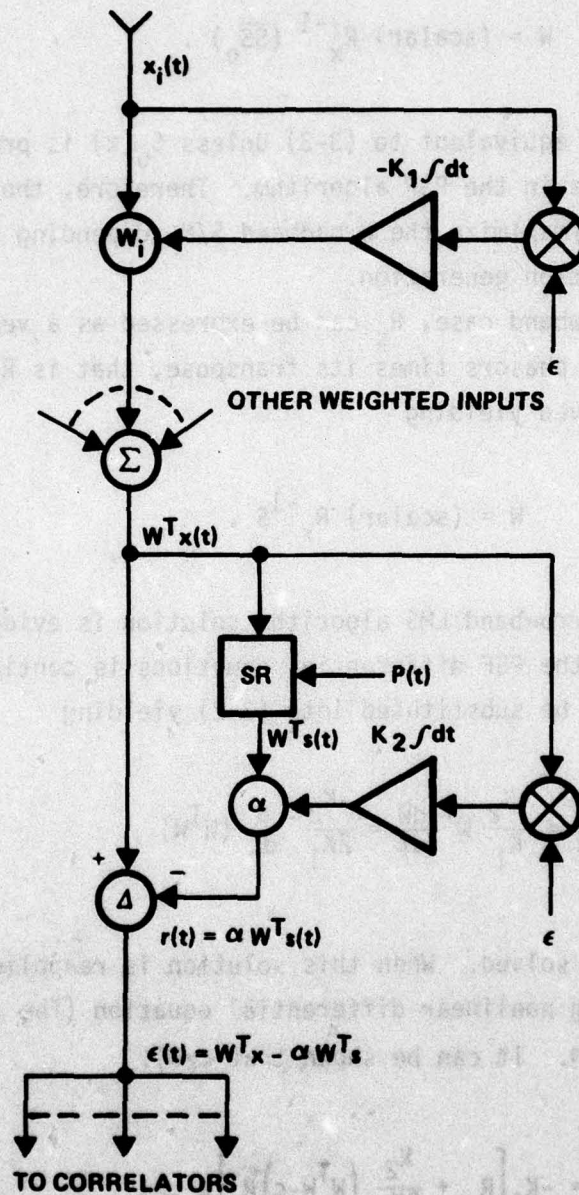


Figure 3.1-1. PSF Algorithm Block Diagram

like the a priori function in a least mean squares sense. The well known weight vector obtained is

$$W = (\text{scalar}) R_x^{-1} (\overline{SS}_0) . \quad 3-4$$

In general, (3-4) is not equivalent to (3-3) unless $S_0(t)$ is proportional to the function $W^T S(t)$ as in the PSF algorithm. Therefore, the LMS algorithm may or may not maximize the broadband S/N, depending upon the method of reference function generation.

In the narrowband case, R_s can be expressed as a vector composed of incident signal phasors times its transpose, that is $R_s = SS^T$. Then (3-3) is easily solved yielding

$$W = (\text{scalar}) R_x^{-1} S . \quad 3-5$$

The similarity to the narrowband LMS algorithm solution is evident.

Solution of the PSF differential equations is continued by observing that (3-1) can be substituted into (3-2) yielding

$$\frac{d\alpha}{dt} = \frac{-K_2}{K_1} W^T \frac{dW}{dt} = \frac{-K_2}{2K_1} \frac{d}{dt} (W^T W) . \quad 3-6$$

Equation (3-6) is easily solved. When this solution is reapplied to (3-1), one obtains the following nonlinear differential equation (The term c is a constant of integration. It can be shown that $c \geq 0$).

$$\frac{dW}{dt} = -K_1 \left[R_x + \frac{K_2}{2K_1} (W^T W - c) R_s \right] W . \quad 3-7$$

Briefly, the eigenvalues and eigenvectors of the matrix terms in (3-7) are not constant, and a coordinate transformation at first does

not appear useful in obtaining a solution. However, since R_x is the sum of R_s and the noise crosscorrelation matrix R_n , it is evident that (3-7) may be regarded as having constant noise terms plus signal terms with a variable power coefficient. From explicit eigenvalue and eigenvector analysis of the crosscorrelation matrix for the case of thermal noise, a single jammer, and a single signal it is possible to approximately diagonalize and subsequently solve (3-7) given that the inner product of the signal and jammer steering vectors is small and that the resulting signal associated eigenvalue is not approximately equal to the jammer associated eigenvalue (i.e., that the coefficient of R_s in (3-7) does not make the effective signal power approximately equal to the jammer power).

In the transformed weight domain, PSF solutions for the thermal noise and jammer associated weights are equal to those for the LMS algorithm given the same angle of arrival and relative power restrictions. One weight, that associated with the signal, is substantially different. The well known LMS solution for the transformed signal associated weight in this special case is

$$u_s = c_{s0} e^{-K_1 [\sigma^2 + S^T S] t} + \left(1 - e^{-K_1 [\sigma^2 + S^T S] t} \right) u_{s0} \quad 3-8$$

Comparatively, the approximate transformed PSF signal associated weight is

$$u_s = \left\{ \frac{\left[\sigma^2 + S^T S \left| \frac{K_2}{2K_1} (c - c_e) - 1 \right| \right]^{1/2}}{\left[\frac{K_2}{2K_1} S^T S + c_s^2 e^{2At} \right]^{1/2}} \right\} \quad 3-9$$

where c_s^2 is a constant of integration, c_e is an approximation of the sum of the squares of the other transformed weights which decay either much

slower or much faster than the exponential in (3-9), and where the symbol A is as follows:

$$A = K_1 \left\{ \sigma^2 + S^T S \left[1 - \frac{K_2}{2K_1} (c - c_e) \right] \right\} \quad 3-10$$

Note that the sign of A is crucial in governing the action of the algorithm. If loop gain K_2 or signal power is too small, then u_s may approach zero. Conversely, for sufficiently large loop gain, A is negative and the exponential in (3-9) decreases with time; thus in the limit u_s approaches a constant. The analogy between PSF algorithm signal adaptation and the transient buildup of an oscillator as a function of loop gain is striking.

Note that the signal adaptation time constant can be adjusted (via K_2) with respect to that of the jamming, but such higher rate of adaptation adjustment can be maintained only as long as output S/N is less than unity. Past this threshold, LMS-type signal adaptation is seen.

As described the PSF algorithm has one more weight than is required theoretically. This is because the additional weight α provides for arbitrary scaling of the input weights without change in the output signal-to-noise ratio. A very useful weight constraint, one which does not affect the optimum solution, is that of fixing one of the input weights. This expedient also eliminates the possibility of failure to adapt for weak signals or small K_2 since a zero weight vector is no longer possible. Furthermore, since the variable input weights adjust with respect to the fixed weight, an optimum pattern is obtained without the need for weight "AGC" action as in the LMS algorithm with its constant reference function amplitude.

Consider the fixed weight constraint during signal acquisition. If a desired signal is not present, output of the signal recognizer is uncorrelated noise, and thus, the array weights are tending to suppress

all signals in the environment. Without a fixed weight constraint, this suppression would result in the algorithm turning off all of the array weights as is the case of the LMS algorithm. Since the fixed weight PSF algorithm cannot have a weight vector solution of zero, the array seeks a minimum solution by acting to null all interferences which are present.

Meanwhile, the alpha loop is caused to respond so that alpha is set at its maximum value in anticipation of a desired signal appearing. When such a desired signal does appear it is likely that substantial pattern gain toward the desired signal exists due to the resultant suppression pattern.

The fact that the array has pre-suppressed the jammers and almost invariably has useful gain toward the desired signal at the instant that the signal appears is a very important system benefit of the PSF algorithm. As time progresses after appearance of the desired signal, the alpha loop causes a refinement of the desired signal gain so as to maximize signal-to-noise ratio, completing the adaptation transient of the algorithm.

In comparison, the LMS algorithm would have to start from a weight vector solution of zero, trying to establish nulls on the interferences while simultaneously trying to bring up gain on the desired signal. If the desired signal is relatively short in duration, the LMS array might never obtain a useful solution.

The PSF algorithm nonlinear differential equation has been analyzed for stability on another study,* and it has been found that all of the eigenvectors of (3-3) which do not correspond to the maximum signal to noise ratio case are unstable. Furthermore, if one unique maximum S/N case exists, this solution is stable and, consequently, is the one the algorithm attains. It is noted that if repeated exactly equal maximum S/N solutions exist then the PSF algorithm might demonstrate an instability by transitioning between these equivalent solutions. Although such behavior has not been seen experimentally or during computer simulation, this point should be resolved analytically in the future.

*Adaptive Array Techniques, Contract No. F30602-78-C-0019 for RADC.

3.1.2 Filtered Steering Vector PSF

It is noted that this variation of the PSF was developed on another contract which is still in progress.* Therefore, only a summary of the algorithm's properties is given here.

As shown in Section 3.1.1, the parameter α of the PSF algorithm is equal to the array output noise to signal power ratio. Therefore, when output signal-to-noise ratio is poor, α is large. Additionally, it is sometimes desirable to utilize a time delay signal recognizer so that the uncorrelated noise term present has essentially the same power as the noise output from the array. When this term is multiplied by the large alpha, the resultant array error signal, although it results in the correct expected value correlations, can result in substantial weight jitter if the adaptive loop gain is high. In some applications, this jitter does not cause a significant problem because signal-to-noise ratio rapidly improves as the array adapts.

In the applications seen for the results of this Study, the need for the Filtered Steering Vector PSF is marginal since signal to noise ratio at the receiver's output needs to be relatively high for acceptable performance. This is due, of course, to the fact that any spread spectrum processing gain was realized by despread at the receiver's input and subsequent receiver bandpass filtering; no further signal processing is available. Given that a useful output is obtained, then it can be concluded that the signal recognizer noise term $n(t)$ is negligible. However, with the extra weight jitter noise present in a fast adapting perturbational algorithm, it was decided that any simple practical measures which would serve to reduce weight fluctuation noise would be desirable.

Let us briefly consider the basic PSF error voltage for a time delay signal recognizer. We have

$$\epsilon(t) = W^T x(t) - \alpha [W^T S(t) + W^T n(t)] \quad 3-11$$

*Adaptive Array Techniques, Contract No. F30602-78-C-0019 for RADC.

where $n(t)$ is the signal recognizer uncorrelated noise term. Clearly, if α is large, the uncorrelated noise term dominates $\epsilon(t)$.

A block diagram of the modified PSF algorithm is given in Figure 3.1-2. Observe that the signal recognizer output is now connected at unity gain to the difference port forming an error term $\epsilon_n(t)$. We get

$$\epsilon_n(t) = W^T x - [W^T S + W^T n] . \quad 3-12$$

This term is referred to as the array noise error. The reason for this identification is evident when (3-12) is expanded. We get

$$\epsilon_n(t) = W^T n + W^T S - [W^T S + W^T n] = W^T n - W^T n . \quad 3-13$$

As can be seen this error term is devoid of signal. Furthermore, since the "time delay" signal recognizer uncorrelated noise term has the same power as $n(t)$, ϵ_n is only 3 dB more noisy than $\epsilon(t)$ for an ideal array. Such a noise increase leads to a comparatively trivial weight jitter increase.

Returning to the block diagram, we see that the signal recognizer output is multiplied by β yielding the signal error term ϵ_s . (We show later that $\beta = \alpha - 1$.) The terms ϵ_n and ϵ_s are distributed to correlators at both the array level and at the beta loop level to obtain the correlation values c_n and c_s . The signal correlation terms, of course, form the signal steering vector. This term is noisy due to the presence of the noise term $n(t)$ in ϵ_s , but these zero mean fluctuations are largely removed by the lowpass filter which follows, thus the input to the integrator c_x is the difference between the direct suppression term and the averaged signal steering vector.

The reasons for these several connections and a proof that the resultant algorithm remains the optimum PSF follows. The resultant modified PSF expected value differential equations are

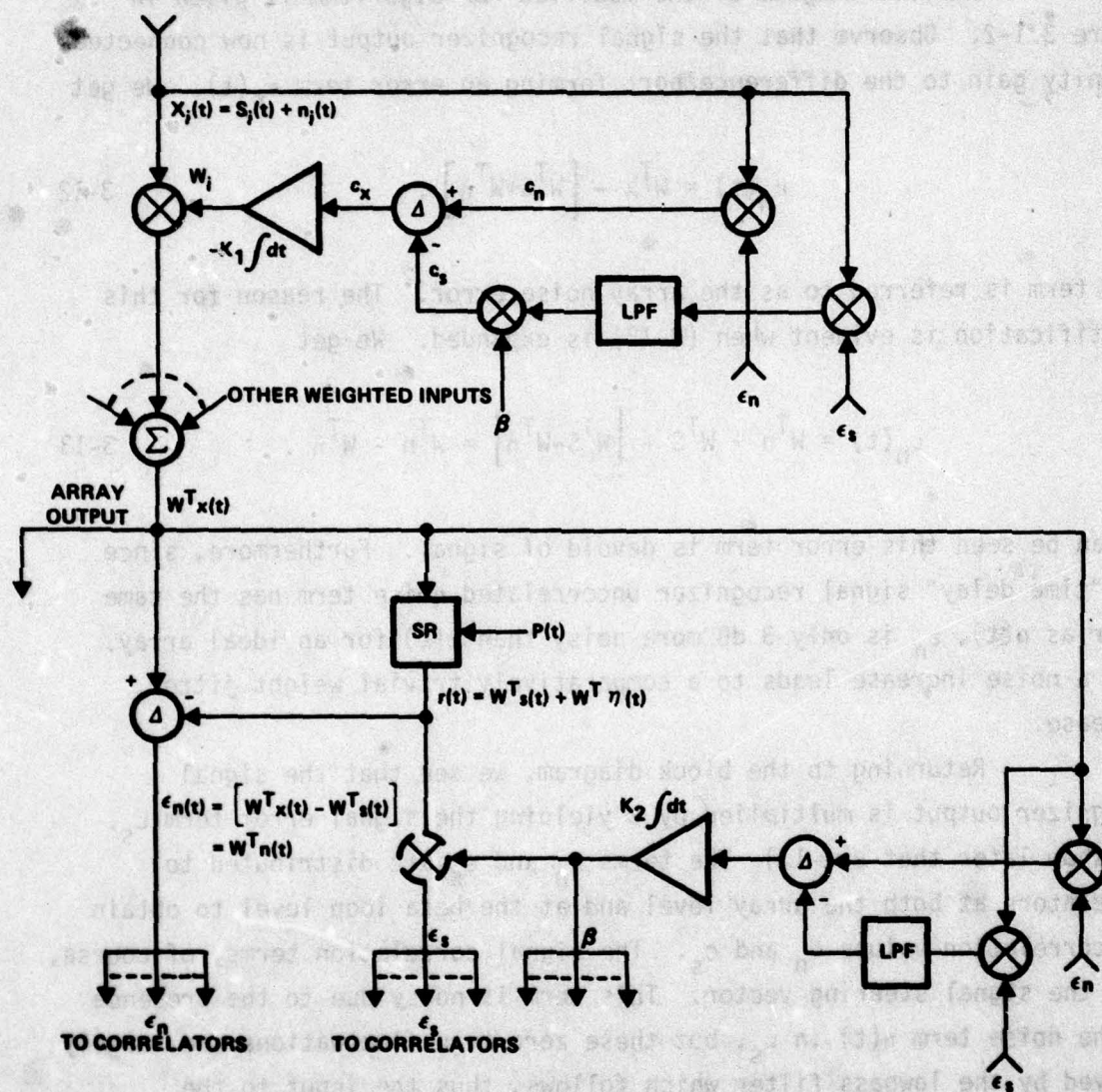


Figure 3.1-2. Filtered Desired Signal Steering Vector PSF

$$\dot{W} = -K_1 [R_n W - \beta \{h(t) * v_s(t)\}] \quad 3-14$$

and also

$$\dot{\beta} = K_2 [W^T R_n W - W^T \{h(t) * v_s(t)\}] \quad 3-15$$

where

$$v_s(t) = [x(t)S^T(t) + x(t)n^T(t)]W. \quad 3-16$$

Concerning the adaptation transient of the system of differential equations, it can be seen that the noise terms are directly suppressed. On the other hand, there is a delay in application of the desired signal steering vector terms due to the convolution of these terms with the lowpass filter impulse response function $h(t)$.

It is important to note first that the algorithm has no negative feedback for the desired signal terms since it has signal steering information available only in the lowpass filtered term. This means that the algorithm adapts by first nulling the interference then placing enhanced pattern gain the direction of the desired signal via the signal steering vector term. In a case where jammer power substantially dominates desired signal power, signal response eigenvalues are relatively small, and according to simulation runs, are more important in determining the signal gain response time than the effects of the lowpass filter.

The reason for the unity gain connection between the signal recognizer and the array output difference port can now be made clear. In the absence of this connection, the differential equation governing the array weights would be

$$\dot{W} = -K_1 [R_n W + R_s W - \beta \{h(t) * v_s(t)\}] \quad 3-17$$

where we have used the fact that $R_x = R_n + R_s$ and we have disregarded fluctuation terms due to x_n . Observe that in this case, the array is acting to suppress both noise and desired signal in a direct manner. Alternatively, it is seeking to restore desired signal gain through the filtered path. Obviously, the two different time constants affecting desired signal gain might cause an instability. (For example, consider the case when only a desired signal is present.) In comparison, the modified PSF differential equations do not have the desired signal suppression term and do not display this instability.

Let us now examine Equations (3-14) and (3-15) for their steady-state solutions. From Equation (3-15), we get

$$\beta_0 = \frac{W^T R_n W}{W^T R_s W} \quad 3-18$$

Similarly, from Equation (3-14), we get

$$R_n W = \beta R_s W \quad 3-19$$

where we have recognized that

$$E\{v_s(t)\} = R_s W \quad 3-20$$

Using the fact that $R_x = R_s + R_n$, (3-18) and (3-3) we calculate the value of β_0 obtained in terms of α . We have

$$\beta_0 = \frac{W^T R_n W}{W^T R_s W} = \frac{W^T R_x W}{W^T R_s W} - 1 = \alpha_0 - 1 \quad 3-21$$

Similarly, for $W=0$ we get

$$R_n W = \beta_0 R_s W . \quad 3-22$$

Adding $R_s W$ to each side of (3-22), using the fact that $R_x = R_n + R_s$, and then substituting the value of α_0 yields

$$R_x W = \alpha_0 R_s W = \left(\frac{W^T R_x W}{W^T R_s W} \right) R_s W . \quad 3-23$$

Thus, steady-state results obtained by the modified PSF algorithm are completely equivalent with those of the basic PSF algorithm. Equally important, weight jitter fluctuations for the modified PSF algorithm can be reduced as desired by appropriate design of the lowpass filter and beta loop gain parameters.

3.1.3 Expanded PSF Algorithm

Consider the problem of simultaneously maximizing the response to several desired signals. We begin by postulating a performance measure to be maximized, then show how such maximization can be achieved using a variation of the PSF algorithm.

Intuitively, we have the performance measure for each of the desired signals (signal-to-noise ratio) which we would like somehow to combine into an overall performance measure which we can maximize. It is readily evident that the sum of performance measures is undesirable since response to weak desired emitters could be omitted entirely in favor of enhancing the response of a single, very powerful desired emitter. Alternatively, if the overall performance measure is regarded as the product of the individual performance measures, then it is evident that weak desired signals cannot be sacrificed for a single strong one. Therefore, we postulate that a global performance measure equal to the product of the individual performance measures is a desirable quantity to maximize. We will find that such a global performance measure leads to a relatively easily realizable algorithm which remains well behaved even in the event that one or more of the desired signals is zero.

Let the array input voltage vector be composed of several desired signals plus noise. That is

$$x(t) = \sum S_i(t) + n(t) . \quad 3-24$$

An individual performance measure for each of these desired signals, γ_i , may be written as follows:

$$\gamma_i = \frac{P_{s_i}}{P_x} \quad 3-25$$

Writing γ_i in terms of the array weight W gives

$$\gamma_i = \frac{W^T R_{s_i} W}{W^T R_x W} = \frac{1}{\alpha_i} . \quad 3-26$$

We now define a global performance measure P . We postulate that P is a desirable performance measure. It is

$$P \equiv (\gamma_1 \gamma_2 \dots \gamma_m) . \quad 3-27$$

Taking the gradient of P with respect to the array weights, then setting the result equal to zero will result in a maximization of the global performance measure. We get

$$\nabla P = P \sum \left(\frac{\nabla \gamma_i}{\gamma_i} \right) = P \sum \alpha_i \nabla \gamma_i \quad 3-28$$

Where we have used the fact that

$$d(uv) = u dv + v du = uv \left[\left(\frac{dv}{v} \right) + \left(\frac{du}{u} \right) \right]. \quad 3-29$$

Calculating the gradient of an individual signal-to-noise ratio yields

$$\nabla_W(\gamma_i) = \nabla \left(\frac{W^T R_{S_i} W}{W^T R_X W} \right) = \frac{-2\gamma_i}{W^T R_X W} \left[R_X - \alpha_i R_{S_i} \right] W \quad 3-30$$

where we have again used the fact that $\gamma_i = 1/\alpha_i$ when appropriate. Summing these terms to yield the gradient of the global performance measure yields

$$\nabla P = \frac{-2P}{(W^T R_X W)} \left[m R_X - \sum \alpha_i R_{S_i} \right] W \quad 3-31$$

where we have let m be the number of simultaneous desired signals.

It is evident that the factor $\frac{2P}{W^T R_X W}$ is merely an adaptation gain factor. Since this term cannot contribute to the optimum results obtained, affecting only algorithm adaptation speed, we will discard this term in favor of a constant gain parameter K . Using the form of the basic PSF differential equation as a guide, we set the rate of change of the adaptive weights equal to a negative constant times the gradient of the global performance measure. Calculation of the individual emitter signal-to-noise ratios, α_i , is conveniently done with individual alpha adaptation loops, each with its own signal recognizer. Consequently, the algorithm differential equations are

$$\dot{W} = -K_1 \left[m R_X - \sum \alpha_i R_{S_i} \right] W \quad 3-32$$

and

$$\dot{\alpha}_i = K_2 W^T \left[R_X - \alpha_i R_{S_i} \right] W \quad 3-33$$

It is important to note that independent maximization of the several desired signals requires an independent a priori discriminant for each of these. If the signals are not independent, then it is evident that a composite waveform can be found so as to define an equivalent single desired term.

We emphasize that a simple summing of several desired signal steering vectors is incorrect and will not result in a maximization of the several desired signals simultaneously. The reason is that a simple summing of steering vectors can easily be shown to be equivalent to the summing of coherent desired signals. Naturally, this means that there is an equivalent single coherent signal; therefore, the array will be "optimally" steered to receive this non-existent sum signal which may even have an imaginary direction of arrival. Since the independent desired signals were not in fact coherent and cannot be summed to form an equivalent signal, the result will be non-optimum.

A block diagram illustrating the expanded PSF differential equations is shown in Figure 3.1-3. Observe that the array correlator and weight control circuitry is the same as for the conventional PSF algorithm. Furthermore, note that independent signal-to-noise ratio tracking loops are provided for each desired signal. In the event that fewer desired signals than tracking loops are present, those signal recognizers which are unused will simply provide uncorrelated noise as outputs. This uncorrelated noise will not affect the algorithm's expected value result but will result in that signal's alpha loop being set to maximum gain to provide rapid adaptation if that signal should appear in the environment. Thus, the expanded PSF reduces to the case of the basic PSF if only one desired signal is present. Also notice that the parameter α_j will automatically adjust to accommodate the parameter m as the number of desired signals in the environment changes.

3.2 Perturbational Correlation

As was noted in Chapter 1.0, there are important economic as well as size, weight and power benefits to be gained through the utilization of a single receiver connected to the adaptive antenna.

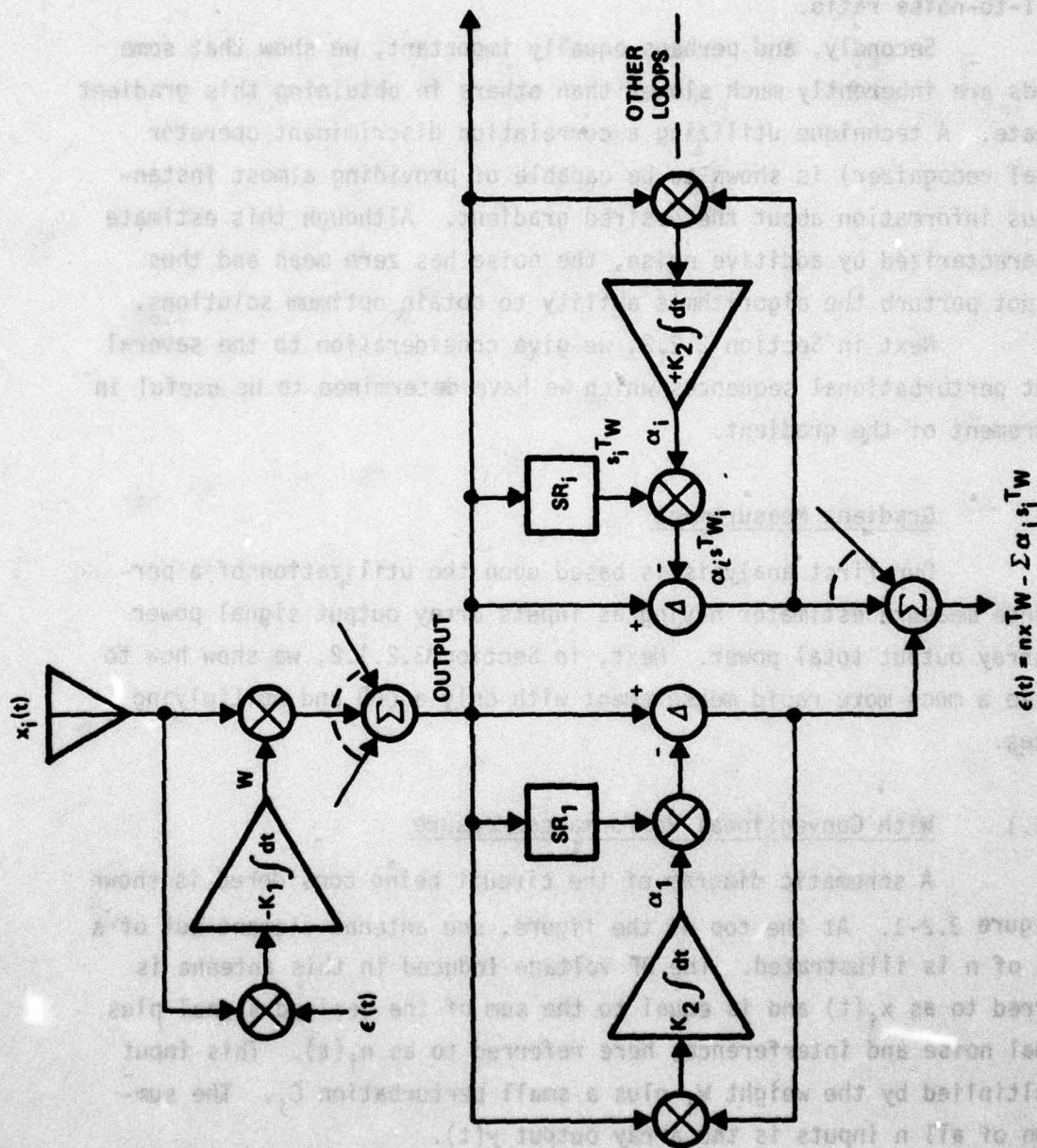


Figure 3.1-3. PSF Algorithm for Multiple Simultaneous Desired Signals

The primary goal of this section is to show how these several adaptive weights can be perturbed so as to enable a measurement of the gradient of the error surface desired, in this case the gradient of the output signal-to-noise ratio.

Secondly, and perhaps equally important, we show that some methods are inherently much slower than others in obtaining this gradient estimate. A technique utilizing a correlation discriminant operator (signal recognizer) is shown to be capable of providing almost instantaneous information about the desired gradient. Although this estimate is characterized by additive noise, the noise has zero mean and thus does not perturb the algorithm's ability to obtain optimum solutions.

Next in Section 3.2.2, we give consideration to the several weight perturbational sequences which we have determined to be useful in measurement of the gradient.

3.2.1 Gradient Measurement

Our first analysis is based upon the utilization of a performance measure estimator having as inputs array output signal power and array output total power. Next, in Section 3.2.1.2, we show how to realize a much more rapid measurement with only a CDO and multiplying devices.

3.2.1.1 With Conventional Performance Measure

A schematic diagram of the circuit being considered is shown in Figure 3.2-1. At the top of the figure, one antenna element out of a total of n is illustrated. The RF voltage induced in this antenna is referred to as $x_i(t)$ and is equal to the sum of the desired signal plus thermal noise and interference, here referred to as $n_i(t)$. This input is multiplied by the weight W_i plus a small perturbation C_i . The summation of all n inputs is the array output $y(t)$.

While implementation details differ among the several realizations of performance measures the block diagram shown next on the figure is representative. Array output $y(t)$ is directed to a power

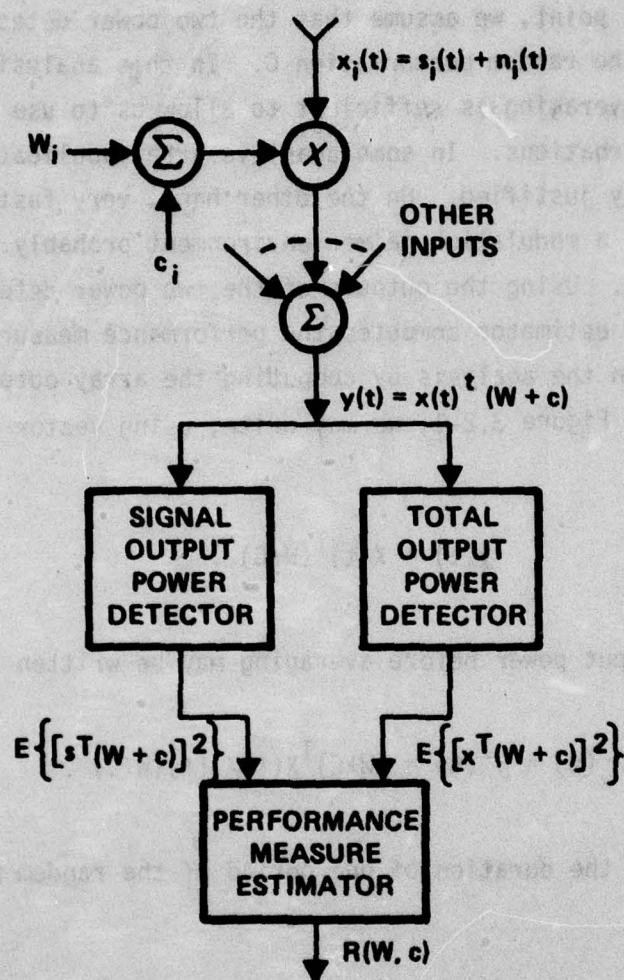


Figure 3.2-1. Perturbational Algorithm Having A Conventional Performance Measure Estimator

detector which measures the total array output. Additionally, it is directed to a power detector which detects only desired signals (a signal recognizer can be used in this device).

At this point, we assume that the two power detectors average for the period of the random perturbation C . In this analysis, it is assumed that this averaging is sufficient to allow us to use expected values of the perturbations. In some adaptive array applications, this assumption is easily justified. On the other hand, very fast adapting arrays operating in a modulation jammer environment probably do not meet this condition. Using the outputs of the two power detectors, a performance measure estimator computes the performance measure $R(W, C)$.

We begin the analysis by computing the array output $y(t)$. From examination of Figure 3.2-1, we may write, using vector arithmetic, that

$$y(t) = x(t)^T(W+C) . \quad 3-34$$

The total array output power before averaging may be written as

$$P_x(t) = y^2(t) = (W+C)^T x(t) x^T(t) (W+C) . \quad 3-35$$

After averaging for the duration of one period of the random parameter C , we get

$$E(P_x(t)) = \bar{P}_x = (W+C)^T R_x (W+C) . \quad 3-36$$

In order to obtain (3-36) one needs to assume that W and C are quasi-static over the integration time and that the variable x is a radio frequency quantity which varies very rapidly.

Desired signal output power is obtained in the same manner as (3-36). We get

$$E(P_s(t)) = \bar{P}_s = (W+C)^T R_s (W+C) . \quad 3-37$$

Naturally, a variety of performance measures can be generated. One of the most useful is the ratio of signal to signal-plus-noise. If we set received signal power constant as in a mainbeam constrained optimization algorithm, we can also use this performance measure in an analysis of that type of algorithm. The ratio of signal to signal-plus-noise is computed by dividing Equation (3-36) into Equation (3-37). We get

$$R(W,C) = \frac{(W+C)^T R_s (W+C)}{(W+C)^T R_x (W+C)} \quad 3-38$$

Expansion of (3-38) gives

$$R(W,C) = \frac{W^T R_s W + C^T R_s C + 2C^T R_s W}{W^T R_x W + C^T R_x C + 2C^T R_x W} . \quad 3-39$$

In order that the perturbation C not introduce excessive adaptive array performance degradation through the mechanism of weight jitter, the perturbation is generally kept small compared with W . For the purpose of this analysis, we will assume that this is so since such an assumption greatly simplifies the analysis. If desirable for a future analysis, a series expansion of higher order terms is feasible but it is not expected that dramatically changed results would be obtained.

According to the assumption that C is small, the terms containing C^2 in Equation (3-39), $C^T R_s C$ and $C^T R_x C$, are second order and are therefore negligible in a first-order analysis. Equation (3-39) is then simplified as follows:

$$R(W,C) \approx \frac{W^T R_S W}{W^T R_X W} \left[\frac{1 + \frac{2C^T R_S W}{W^T R_S W}}{1 + \frac{2C^T R_X W}{W^T R_X W}} \right]. \quad 3-40$$

Again, we use the fact that the perturbation C is a small first-order variation; consequently, the denominator of (3-40) may be rewritten approximately as follows

$$R(W,C) \approx \frac{W^T R_S W}{W^T R_X W} \left[1 + 2 \frac{C^T R_S W}{W^T R_S W} \right] \left[1 - 2 \frac{C^T R_X W}{W^T R_X W} \right]. \quad 3-41$$

Expansion of (3-41) yields four terms. The product of the two small terms containing C is seen to be second-order in C and therefore is negligible in a first-order analysis. Consequently, we get

$$R(W,C) \approx \frac{W^T R_S W}{W^T R_X W} \left[1 + \frac{2C^T R_S W}{W^T R_S W} - \frac{2C^T R_X W}{W^T R_X W} \right]. \quad 3-42$$

Equation (3-42) is expanded one step further by distributing the coefficient of the brackets then factoring $2C^T$. Observe that this coefficient is simply the expected value of the signal to signal-plus-noise ratio. For convenience in notation, we shall call this ratio Q (for Quality), we have

$$\frac{P_s}{P_x} = \frac{P_s}{P_s + P_n} = \frac{W^T R_s W}{W^T R_x W} = Q. \quad 3-43$$

Using (3-43) in (3-42) yields

$$R(W, C) = \frac{W^T R_s W}{W^T R_x W} + 2C^T \left[\frac{R_s W}{W^T R_x W} - \frac{(W^T R_s W)}{(W^T R_x W)^2} R_x W \right]. \quad 3-44$$

We note that twice the term in brackets in Equation (3-44) is exactly the gradient of the signal to signal-plus-noise ratio. We will show this fact. We begin by taking the gradient of Q , where the gradient operator ∇ is defined as follows: (the u_i are unit vectors in the "n" dimensional space of W).

$$\nabla_W = u_1 \frac{\partial}{\partial W_1} + \dots + u_n \frac{\partial}{\partial W_n}. \quad 3-45$$

Calculating the gradient of Q yields

$$\nabla_W(Q) = \frac{1}{W^T R_x W} \nabla(W^T R_s W) - (W^T R_x W)^{-1} \nabla(W^T R_x W). \quad 3-46$$

Detailed evaluation of the partial derivatives gives

$$\nabla_W(Q) = 2 \left[\frac{R_s W}{(W^T R_x W)} - \frac{(W^T R_s W)}{(W^T R_x W)^2} R_x W \right]. \quad 3-47$$

That the bracketed terms in Equations (3-44) and (3-47) are equal is readily seen.

When (3-47) is substituted into (3-44), we obtain the first-order approximation of the performance measure in a form suitable for our use. One gets

$$R(W,C) = \left(\frac{W^T R_S W}{W^T R_X W} \right) + C^T \nabla_W \left(\frac{W^T R_S W}{W^T R_X W} \right) = Q + C^T \nabla_W Q \quad 3-48$$

A brief interpretation of (3-48) is in order. The expected value of the performance measure at the end of one period of the perturbation C is equal to the present value of signal to signal-plus-noise ratio for the current value of the weight vector W plus a small perturbation term which is the projection of the vector C on the gradient of the S/N ratio. The magnitude of the perturbation is largely dependent upon how nearly the vector C is aligned with the gradient. Large perturbations indicate that C is practically colinear with the gradient while zero perturbations indicate orthogonality. Similarly, large negative perturbations indicate that C is approaching an antiparallel alignment with the gradient.

We now demonstrate how an algorithm to maximize the receiver output signal-to-noise ratio can be based upon Equation (3-48). Consider the following weight adjustment procedure.

$$W_{i+1} = W_i - K C_i R(W, C_i) \quad 3-49$$

Note that the above is a vector equation because we are multiplying the vector of weight perturbations C times the scalar performance measure parameter. This quantity, when subtracted from the present weight value W_i , becomes the new weight value W_{i+1} . We calculate the change in W by subtracting W_i from each side of the above equation, then expand it by substituting for $R(W,C)$ we get

$$\Delta W_1 = -K[CQ + CC^T \nabla Q] \quad 3-50$$

Observe that if we had the expected value of ΔW equal to the gradient of the signal-to-noise ratio, we would have the differential equation we desire. We will do an expected value operation on (3-50), then determine the properties C must have in order to yield the desired result. We get

$$E(\Delta W) = -K[E(CQ) + E(CC^T \nabla Q)] \quad 3-51$$

where the expected value operation is defined as

$$E(x) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} x(t) dt \quad 3-52$$

It is important to recognize that we want to hold W constant when this expected value operation is performed because we wish to compute the expected value of weight change ΔW for any given, but arbitrary value of W . We may do this conceptually and mathematically by holding W fixed at any possible value W might assume then calculating the expected value of weight change at each of these points.

Since W is fixed, the parameter Q is also fixed as is ∇Q . Consequently, (3-52) may be written as

$$E(\Delta W) = -K[QE(C) + E(CC^T) \nabla Q] \quad 3-53$$

Obviously, the perturbation parameter C should have zero mean in order to cause the first term of (3-53) to vanish. That is,

$$E(C) = 0 \quad 3-54$$

Additional insight is provided by expanding the matrix CC^T . When the indicated expected value is taken, we get

$$E(CC^T) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \begin{bmatrix} c_1^2(t) & c_1(t)c_2(t) & \dots & c_1(t)c_n(t) \\ c_1(t)c_2(t) & c_2^2(t) & \dots & c_2(t)c_n(t) \\ \vdots & \vdots & \ddots & \vdots \\ c_1(t)c_n(t) & c_2(t)c_n(t) & \dots & c_n^2(t) \end{bmatrix} dt \quad 3-55$$

If the vector components of $C(t)$ are uncorrelated with each other, then the expected value of the off diagonal terms of (3-55) will be zero. The diagonal terms, of course, are the expected value of a given sequence squared, i.e., its average power. If we choose all of the perturbational sequences to have equal power, let this value be unity, then the above integral reduces the identity matrix. We get

$$E(CC^T) = I. \quad 3-56$$

Therefore using zero mean uncorrelated sequences results in the expected value of weight change being proportional to the gradient of the signal-to-noise ratio as desired. Using (3-56) and (3-54) in (3-53) gives

$$E(\Delta W) = -K \nabla Q. \quad 3-57$$

In summary, we have determined that if the perturbational vector sequence C has components that are zero mean and time orthogonal to one another, then we can extract the gradient of the performance measure and use the gradient quantity to optimize the adaptive array weights.

We will discuss the possibilities for perturbational sequences in considerable detail in Section 3.2.2, but first, let us examine the

AD-A068 890

HARRIS CORP MELBOURNE FLA

F/G 17/4

APPLICATION OF A CORRELATION DISCRIMINANT OPERATOR TO PERTURBAT--ETC(U)

APR 79 6 P MARTIN

F30602-77-C-0073

UNCLASSIFIED

RADC-TR-79-44

NL

2 OF 4
ADA
068890



block diagram of a circuit to maximize output signal-to-noise ratio which is based on the procedure just developed.

Consider the circuit illustrated in Figure 3.2-2. The array portions of this figure are identical to the basic one, Figure 3.2-1, but interesting details begin in the receiver output where we show the total power detector and the signal power detector. The total power detector consists simply of a multiplier operating in a squaring mode and, in this case, followed by a lowpass filter. We have assumed a conventional spread spectrum demodulator to extract an estimate of the desired signal which is also squared and filtered. We will return to these circuits in a moment, but, continuing development of the algorithm, the total power and signal power quantities respectively are used to generate the performance measure $R(W,C)$ which is distributed to correlators at each of the array weights. There, this term is multiplied by the individual weight perturbational sequences $c_i(t)$, yielding the gradient term which is integrated to obtain W_i .

Although the algorithm is functional, this particular circuit would display reduced performance for two reasons. First, the signal power detector does not contain an unambiguous estimate of signal power. Some noise will always be present to bias this estimate. Additionally, there is substantial time delay in the lowpass filters and the bandpass filter which directly impact the rate at which this algorithm can adapt.

Consider the timing diagram in Part B of this figure. At time zero, a weight perturbation is applied which produces a receiver intermediate frequency output at time τ_R . Due to the delay of the lowpass filter, a time $\tau_X \approx 1/BW_{LPF}$ passes before the total power detector output is obtained; an even longer wait is necessary before the signal power detector output is obtained. Observe that a performance measure cannot be calculated until both of these measurements are available.

Since this study addressed signals such as found in the HF/VHF/UHF band, we are presumably tracking narrowband signals. Suppose that we do very little averaging and allow the lowpass and bandpass

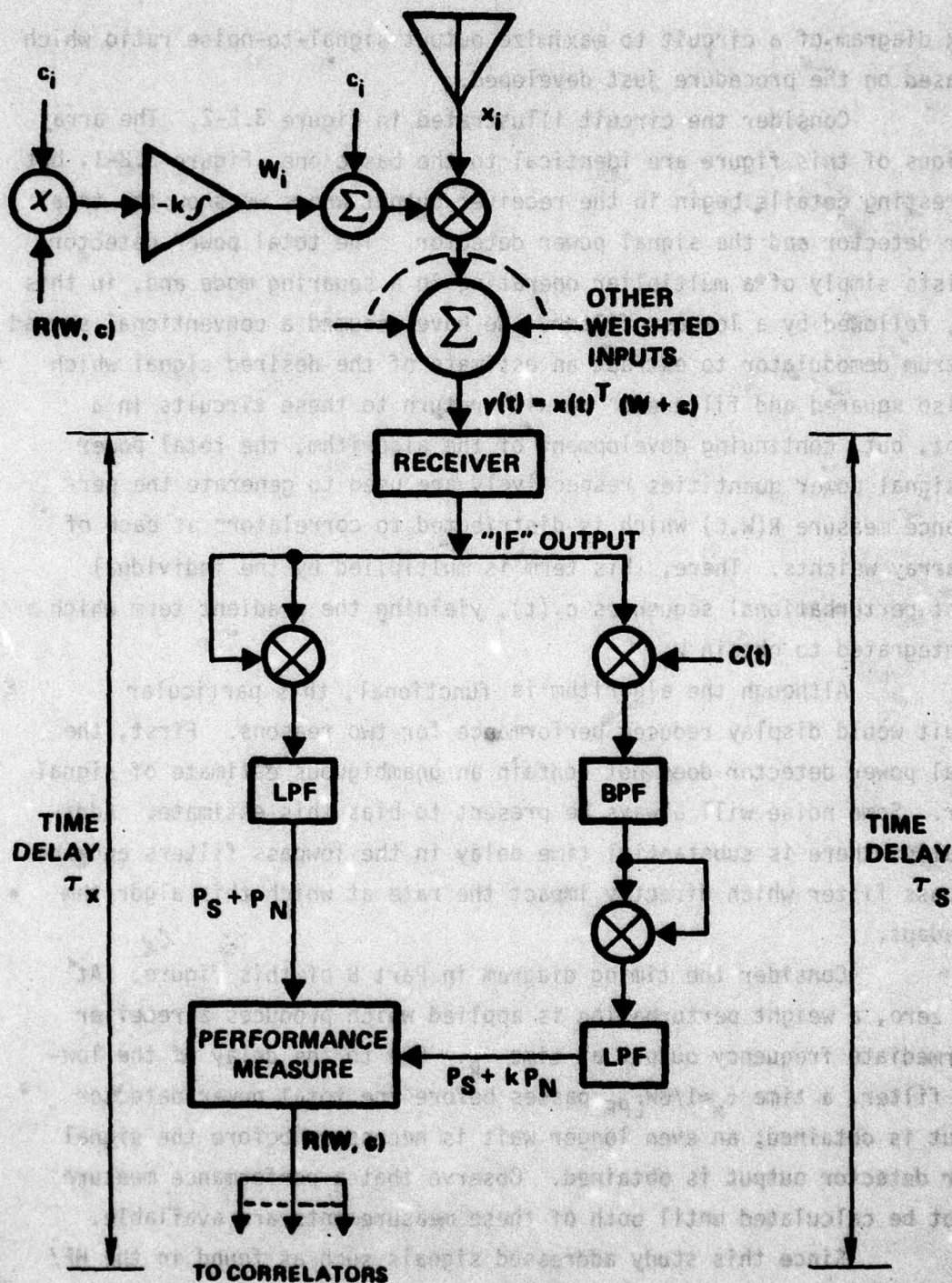


Figure 3.2-2. Perturbational Algorithm Using a Conventional Matched Filter Desired Signal Power Detector

filter to have about 1 kHz bandwidth, then the time delay before the signal power output becomes available is in the order of 2 milliseconds. Considering that 10 to 100 weight iterations are required to obtain 1/e null formation, the corresponding time is 20 milliseconds to 0.2 seconds. This is probably too slow to be useful in most applications. The situation is even worse if more realistic filter bandwidths are assumed, for example, a bandpass filter of 100 Hz. Observe also that the time delay of the receiver itself is trivial compared with these other delays. This is disturbing because theoretically we would like the receiver bandwidth to be the limiting factor in adaptation. (Ultimately one's ability to measure the gradient is dependent upon one's ability to get perturbational information through the receiver.)

3.2.1.2 With CDO and Multipliers

In this section, we illustrate an approach which allows us to extract noisy but unbiased estimates of the gradient components almost instantaneously. Under these conditions, time delay of the receiver becomes the limiting factor in this process. An algorithm based on such a measurement technique would therefore be able to adapt at the theoretically maximum rate.

Consider the block diagram shown in Figure 3.2-3. The array and receiver portion of this figure are exactly the same as in the previous figures. Note that there are two multipliers with the output of the first being labeled P_x and the other P_s . Using the IF output expression given in Equation (3-34), we may write the instantaneous quantity $P_x(t)$; we get

$$P_x(t) = W_{xx}^T W + C_{xx}^T C + 2C_{xx}^T W. \quad 3-58$$

The correlation discriminant operator (CDO) output is

$$r(t) = [W + C(\tau)]^T [S(t) + n(t)] \quad 3-59$$

where $C(\tau)$ is the perturbational sequence delayed by the signal recognizer delay τ . We now assume that τ is negligibly small compared to the perturbational period. Appendix A contains an analysis of the important case obtained when this approximation is not valid.

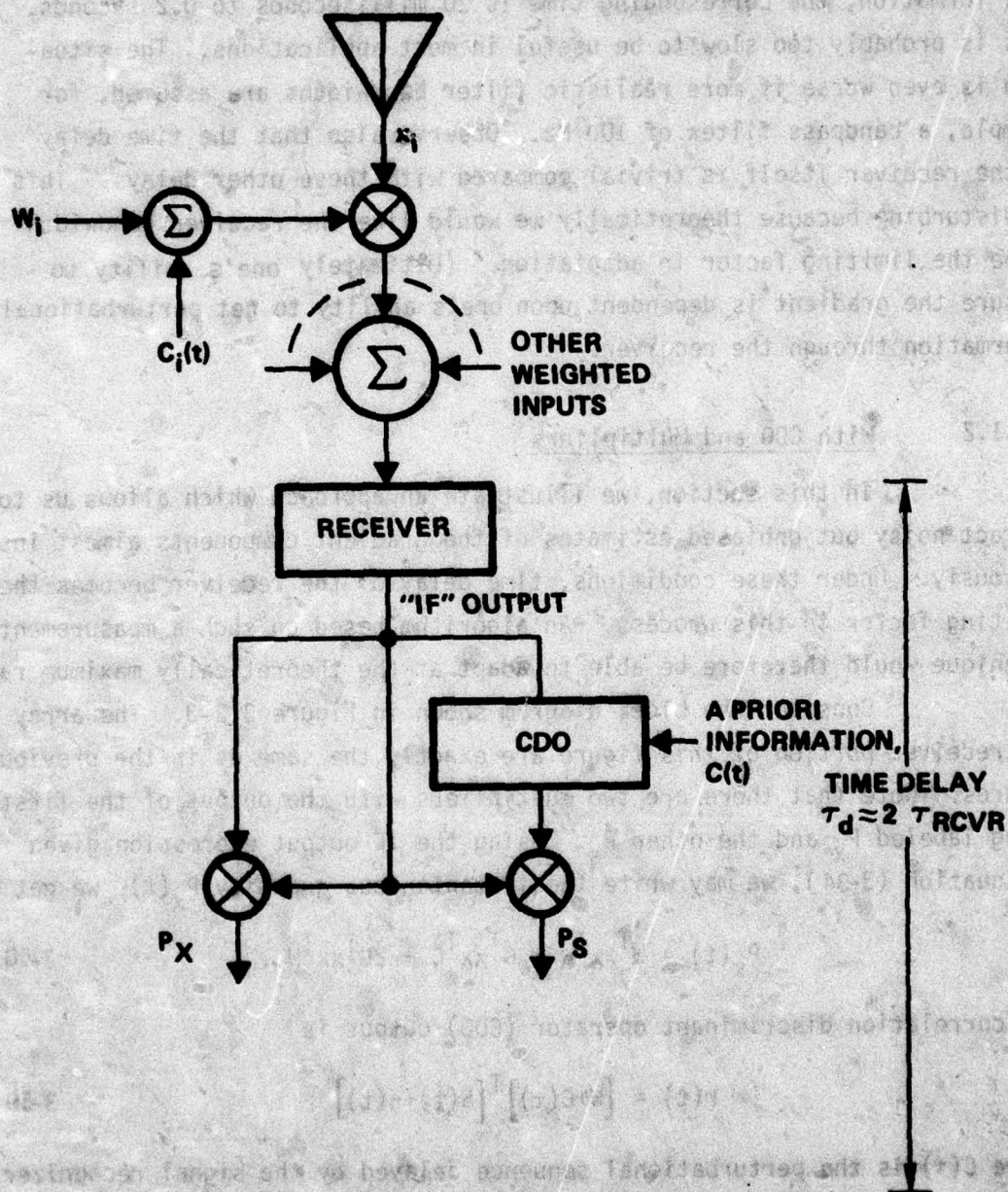


Figure 3.2-3. Perturbational Algorithm Using a Correlation Discriminant Operator for Desired Signal Power Measurement

Computation of $P_s(t)$ yields

$$P_s(t) = (W+C)^T [S(t)+n(t)] [S(t)+n(t)]^T (W+C) . \quad 3-60$$

Expanding the brackets in (3-60) gives

$$P_s(t) = (W+C)^T [SS^T + Sn^T + nS^T + nn^T] (W+C) . \quad 3-61$$

Applying theory of the CDO, we know that the only matrix term in (3-61) having a non-zero expected value is the matrix SS^T . The other terms contribute noise which ultimately causes weight jitter, but they do not contribute a bias. We will lump all of these terms for the time being into one term $N(t)$, then we get

$$P_s = W^T SS^T W + C^T SS^T C + 2C^T SS^T W + N . \quad 3-62$$

Observe that both the term P_s and P_x contain terms having the inner product of the perturbational vector C with precursors of the signal-to-noise ratio gradient components. Respectively these terms are $2C^T SS^T W$ and $2C^T xx^T W$.

We will multiply (3-58) and (3-62) by the vector of perturbational terms $C(t)$ and take expected values. For P_x we get

$$E[C(t)P_x(t)] = E[C(W^T xx^T W) + CC^T xx^T C + 2CC^T xx^T W] . \quad 3-63$$

Similarly, for P_s we find

$$E[C(t)P_s(t)] = E[C(W^T SS^T W) + CC^T SS^T C + 2CC^T SS^T W + CN] . \quad 3-64$$

where we have assumed that W is quasi-static.

Since the perturbation is independent of the inputs $x(t)$, we may immediately simplify (3-63) and (3-64). We get, respectively, that

$$E(CP_x) = E[CC^T R_x C] + 2E[CC^T R_x W] \quad 3-65$$

and

$$E(CP_s) = E[CC^T R_s C] + 2E[CC^T R_s W] \quad 3-66$$

where we have assumed that $C(t)$ has zero mean. Expansion of the terms containing C^3 reveals that all possible combinations of the sequences occur. That is, we get the following products

$$a_c = c_i(t)c_j(t)c_k(t) \quad 3-67$$

where we have used the term a_c to represent the coefficient of terms appearing in the vector due to the perturbational sequences c_i , c_j and c_k . In order to ensure zero expected value of these terms, it is necessary that the perturbational sequences have components that are orthogonal in triplets. This condition is obviously satisfied by random or very long pseudo-random sequences and by certain special cases of orthogonal functions. We discuss this topic in more detail in the next section, Perturbational Sequences.

Applying triplet orthogonality as well as simple orthogonality constraints to (3-65) and (3-66) results respectively in

$$E(CP_x) = 2R_x W \quad 3-68$$

and

$$E(CP_s) = 2R_s W \quad 3-69$$

Observe that Equations (3-68) and (3-69) respectively contain the gradient of the total power and the gradient of the signal power for this array configuration. It is evident, therefore, that these terms can be used to construct a signal maximizing algorithm such as the PSF algorithm. We will demonstrate this procedure in Section 3.3, Modified Algorithms.

3.2.2 Perturbational Sequences

Suitable perturbational sequences for realizing the desired gradient measurements are discussed in the following paragraphs. Random, pseudo-random and two orthogonal sequences are covered. Only the Walsh orthogonal functions are considered to have practical application here, but the other orthogonal set is treated because it very clearly illustrates the nature of the perturbational algorithms.

3.2.2.1 Random Sequences

In the previous section concerning gradient measurement, we established that suitable sequences must have zero mean and must be orthogonal in pairs and triplets. Obviously, truly random sequences satisfy all of these requirements. Observe that we need only a biphase random quantity.

One possible deficiency of random sequences is that they may not converge quickly to the desired zero expected value cross products and therefore might cause undue weight cross coupling.

3.2.2.2 Pseudo-random Sequences

Comments applying to random sequences also apply here, but there is an additional factor of the correlation of one bit out of a total of $(2^m - 1)$ where m is the number of bits used to generate the random sequence. In a practical system, sequences generated from registers having 10 bits or more produce negligible correlation effects for the squared and cubed terms (10^{-6} , 10^{-9} respectively). The mean is in error only by one part in 10^3 .

A particular problem we have encountered with pseudo-random sequences is that they display long runs. During these runs, substantial cross coupling can exist between gradient components. Although this cannot cause a long-term problem, it can cause a short term algorithm performance reduction.

3.2.2.3 Orthogonal Sequences

An important advantage of using orthogonal sequences is that convergence of the outer product of the functions to the identity occurs very rapidly, specifically at the end of an orthogonal sequence. The primary type of orthogonal sequences we have in mind are Walsh functions. However, a different set of orthogonal sequences, illustrated below, provides greater insight into the process that is occurring.

3.2.2.3.1 Three Level Sequence

Consider the following sequences.

$$\begin{matrix} S_1 = \\ S_2 = \\ S_3 = \\ \vdots \\ S_n = \end{matrix} \left\{ \begin{matrix} +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & +1 & -1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & -1 & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & & & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & +1 & -1 \end{matrix} \right\} \quad 3-70$$

Note that the sequences are in fact orthogonal. When a given sequence is +1 or -1 all other sequences are 0. Furthermore, note that the +1/-1 perturbation occurring at a given antenna input specifically extracts an approximation to the partial derivative of the array output with respect to that input. This, of course, is the definition of a gradient measurement process. Therefore, sequential perturbation of the weights, well known in gradient extraction techniques, is a special case of the use of orthogonal sequences for the same purpose. Furthermore, the use of

random variables or pseudo random variables is again the same process except that a longer period of time is required to establish orthogonality and thus unique gradient extraction.

The sequence above is not recommended due to the fact that modulation-type jammers could conceivably be present only during the time that one element's gradient is being measured. This would result in poor adaptation to the modulation jammer. Use of Walsh orthogonal or random sequences ensures that all weights are being perturbed simultaneously permitting better estimates of the gradient necessary to null the modulation jammer.

3.2.2.3.2 Walsh Functions

It is not our purpose to explore Walsh functions in theoretical depth; however, some important factors need to be discussed. Specifically, we will review how the Walsh functions are generated by their recursion relationships. We will then show that products of certain of the Walsh functions are exactly equal to a third Walsh function. This, of course, means that some of the triple sequence products will be totally correlated, and thus, not useful in measurement of the gradient. Fortunately, by eliminating half of the Walsh functions, we can form a set which has the desirable triplet orthogonality property.

The first eight Walsh functions (numbered zero through seven) are illustrated in Figure 3.2-4. Note the strong resemblance some of these functions bear to a "hard limited" sine or cosine wave. On the other hand, note that other of these functions bear no resemblance to sines or cosines.

Further inspection of the sequences will show that they are all of zero mean. Additionally, the product of any two sequences is also a zero mean function.

Now consider the triple product of sequences Number 1, 2, and 3. For convenience, we write these sequences below from left to right. We have

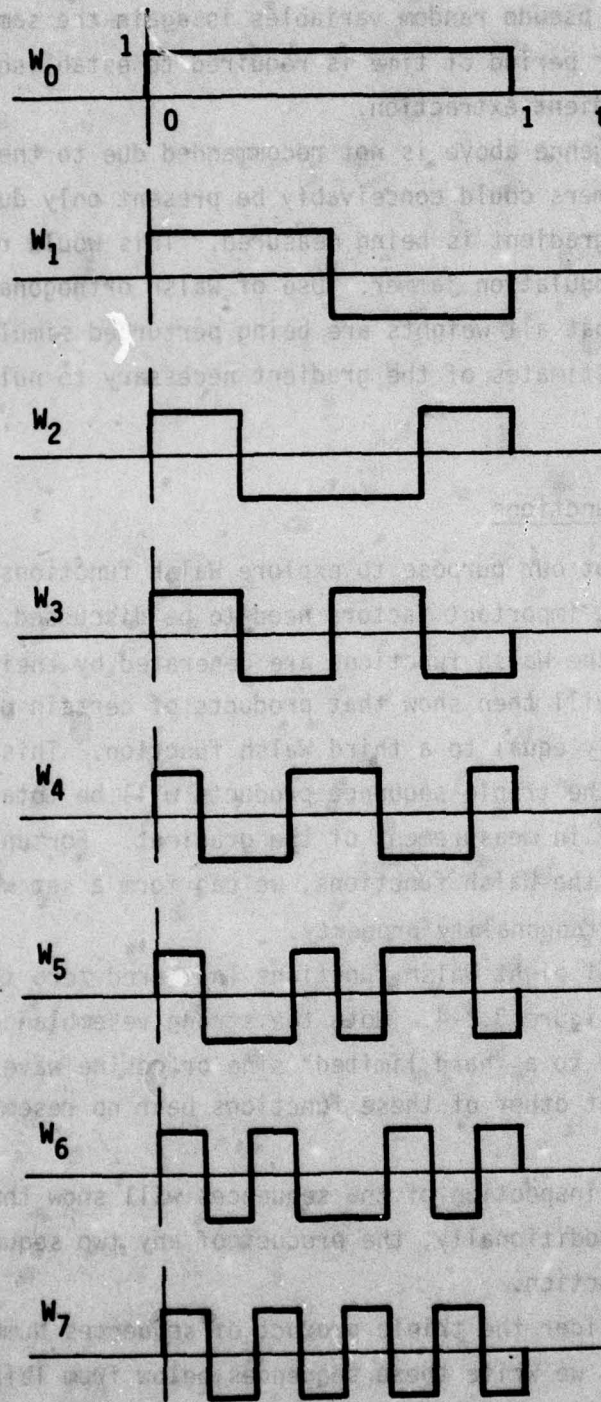


Figure 3.2-4. First Eight Walsh Functions

$$\begin{aligned}
 W_1 &= 1 & 1 & -1 & -1 \\
 W_2 &= 1 & -1 & -1 & 1 \\
 W_3 &= 1 & -1 & 1 & -1
 \end{aligned}
 \tag{3-71}$$

Observe that in any column the product is +1. Therefore, this triplet is totally correlated.

The defining recursion relationship for Walsh functions is given below. We have

$$W_i(t) = W_{k_i}(2t) + (-1)^{k_i + P_i} W_{k_i}(2t-1) \tag{3-72}$$

where $W(t)$ for $t < 0$ or $t > 1$ is zero, i is an integer, P_i is either zero or one, and $i = 2k_i + P_i$. Using (3-72) we can derive an expression for the product of two Walsh functions. We get

$$W_i(t)W_j(t) = W_{k_i}(2t)W_{k_j}(2t) + (-1)^{k_i + k_j + P_i + P_j} W_{k_i}(2t-1)W_{k_j}(2t-1) \tag{3-73}$$

where we have used the fact that $W(2t)$ or $W(2t-1)$ is zero for $2t$ outside the limits of zero to one to eliminate two cross products from the expansion. Using Equation (3-73) in an expansion based on the sequence W_1 , we have been able to derive recursion relationships (following a very tedious procedure) which show that elimination of a certain half of the Walsh functions will guarantee that the remainder are triply orthogonal. We will return to this point in a moment.

Use of the recursion relationships is a relatively difficult way of generating Walsh functions. There is an efficient procedure which produces matrices of the Walsh functions called Hadamard matrices. One begins with a one by one identity matrix (the number 1) and forms a two by two matrix from it by placing it in all quadrants except the lower right hand one. In this last quadrant, the matrix is changed in sign. Reference to Figure 3.2-5 will illustrate this procedure. The next step

SIZE	EQUIVALENCE	EXAMPLE
1x1	$A = 1$	1
2x2	$B = \begin{bmatrix} A & A \\ A & -A \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
4x4	$C = \begin{bmatrix} B & B \\ B & -B \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$

Figure 3.2-5. Generation of Hadamard Matrices

is to take the 2x2 matrix just generated and use it to make a four by four matrix, again with sign unchanged in all but the lower right hand quadrant where the sign of all components of the two by two matrix is reversed.

The columns of the Hadamard matrix are Walsh functions, although their numbering sequence does not correspond to consecutive numbering of the Walsh functions. The first sixteen columns of a 32x32 Hadamard matrix are given in Figure 3.2-6. We can show that the product of any two even numbered columns will be a sequence in some odd number column, and conversely, the product of any two odd numbered columns will be one of the even numbered column sequences.

Consequently, a very simple technique for elimination of the offending sequences is to eliminate every other column. Observe that some columns have very rapidly changing sign while others have very slowly changing sign. In order to ensure maximally rapid weight fluctuation (so as to maximize the probability that all weights will fluctuate during the time a pulse jammer is present) we select the set of sequences varying most rapidly. The selected sixteen sequences from the above 32x32 matrix are given in Figure 3.2-7.

3.2.2.4 FDMA Sequences

In some cases, a receiver connected to the array output will have or can be designed to have a wide enough bandwidth (at least in the first IF stages) to accommodate the perturbation information. If the perturbation sequences are multiplied by a common carrier term $\cos(\omega_p t)$ and ω_p is selected to be greater than twice the desired signal bandwidth, weight perturbations will not degrade the received signal, and can be recovered with a simple correlation technique. This topic is detailed and illustrated in Section 3.3.2.

3.3 Modified Algorithms

In these paragraphs, we address the realization of three algorithms based on the perturbational correlations developed in

	1	2	3	4	5	6	7	8		9	10	11	12	13	14	15	16
1 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
2 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1
3 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
4 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1
5 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
6 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1
7 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
8 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1
9 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
10 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1
11 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
12 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1
13 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
14 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1
15 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
16 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1
17 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
18 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1
19 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
20 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1
21 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
22 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1
23 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
24 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1
25 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
26 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1
27 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
28 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1
29 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
30 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1
31 :	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
32 :	-1	-1	-1	-1	-1	-1	-1	-1		-1	-1	-1	-1	-1	-1	-1	-1

Figure 3.2-7. The Selected Sixteen Sequences from a 32x32 Hadamard Matrix

Section 3.2. A relatively short but important portion is the derivation of the perturbational PSF algorithm which appears in Section 3.3.1. We follow this in 3.3.2 with the derivation of the expected value equivalence of random search with a normalized gradient algorithm. Next is a FDMA perturbational parameter PSF design in 3.3.3. Lastly, in Section 3.3.4, we discuss a simple algorithm based on the logarithm of power changes, also realizing a variation of the PSF algorithm, because this procedure was used in the 8080 microprocessor experiment. Since the principal microprocessor experimentation based on the AMD 2900 device utilized the perturbational PSF described in Section 3.3.1, the log power algorithm work is not emphasized.

3.3.1 Perturbational PSF Algorithm

A simple block diagram of the algorithm is given in Figure 3.3-1. Naturally, the array, receiver, signal recognizer and power detectors are the same as those discussed in Section 3.2.1.2. Since the inner product of the total power and signal power gradient terms is present at the output of the multipliers, the inner product of the noise gradient is conveniently formed by subtracting P_s from the total power term P_x . The signal power term is multiplied by beta so as to facilitate computation of the term $\beta R_s W$ here rather than at each integrator to reduce the need for multiplications. This is permissible if the bandwidth of the beta loop is not too great. We have found through our simulation programs that a satisfactory adaptation rate and satisfactory weight jitter due to the uncorrelated noise term from the signal recognizer result in loop gain parameters which permit the location of the beta multiplier prior to the lowpass filter.

The lower part of the figure illustrates the expansion of the inner product into the desired gradient terms. At each weight, there is a pair of multipliers which correlate the signals respectively from the noise power and signal power ports with the perturbational sequences. Since the sequences are biphasic, these multipliers are actually only sign changers. As we showed in Section 3.2.1, the individual

gradient terms are recovered by these correlations. Therefore, in a vector sense, the weight differential equation is

$$\dot{W} = -K[R_n W - \beta \{h(t) * e_s(t) c(t)\}] \quad 3-74$$

where we have convolved the lowpass filter impulse response $h(t)$ with the desired signal steering vector terms. For simplicity, in terms of expected value parameters and negligible delay effect from the filters, we have the simplified equation

$$\dot{W} = -K[R_n W - \beta R_s W] \quad 3-75$$

Next, we establish the input for the beta loop integrator. This term can be calculated by multiplying the individual W values times the input to the W integrators, then applying the scale factor K_2 we get

$$\dot{\beta} = K_2 [\sum_i W_i (\Delta W_i)] = K_2 W^T [R_n - \beta R_s] W \quad 3-76$$

Interestingly, the dc terms present at P_n and P_s to a first order are $W^T R_x W$ and $W^T R_s W$. The additional second order terms $C^T R_x C$ and $C^T R_s C$ also contribute a dc term which biases the result. However, if the perturbation being added is small, this bias is negligible. Consequently, in most applications, the beta loop integrator can be driven exactly as it is in the basic PSF algorithm form. This considerably simplifies the computation since one does not need to do the multiplications by W . It is interesting that in the majority of simulations reported in Chapter 5.0, we used this second method without any noticeable degradation in results obtained.

Before leaving this topic, we point out that it is possible to combine these perturbational equations so as to yield the term $x^T x$. Given this "total antenna power input" term, one can adjust the algorithm adaptation time constant so as to provide essentially constant adaptation

time as a function of input signal and jammer power changes due to range variations, multipath variations and unexpectedly large jammers. However, the coefficient of $x^T x$ contains the amplitude of the perturbation parameter cubed and therefore is ordinarily a very weak and noisy estimate. Consequently, we have not used this term in our simulations. However, in some other application where relatively large perturbations are considered, the utility of this function should not be overlooked.

3.3.2 FDMA of Search Parameters

The circuit shown in Figure 3.2-3 is unnecessarily subjected to cross correlation noise because the perturbation parameters are located within the same bandwidth as the desired signal. Additionally, array output S/N ratio is degraded by the presence of those same search parameters. In order to solve this dual problem, the circuit shown in Figure 3.3-2 is suggested. Basically, this circuit is the same as the one just discussed and presented in Figure 3.3-1. The exceptional difference is that the search parameters are modulated by a common carrier term $\cos(\omega_p t)$. By design, we will place the frequency ω_p so that the perturbation terms fall outside the bandwidth of the desired signal. In order to accommodate these terms, the receiver (error channel amplifiers) must have at least twice the bandwidth in its IF stages as previously required. The out of band perturbational terms are eliminated from the receiver output by narrowband filtering to the desired signal's bandwidth. Similarly, the perturbation terms are also narrowband filtered by a different filter so as to eliminate the undesired array output terms. Then the array output is used precisely as it is in a conventional PSF algorithm. A conventional error term is also formed.

When the error term is multiplied with the composite perturbational terms we obtain the desired inner product of the perturbational vector with the gradient. This scalar is distributed to array correlators which now multiply the individual $p_i(t)$ terms by the composite term in direct analogy with the RF correlation process. Due to decorrelation of $p_i(t)$ with $p_j(t)$, only the desired gradient is extracted at each

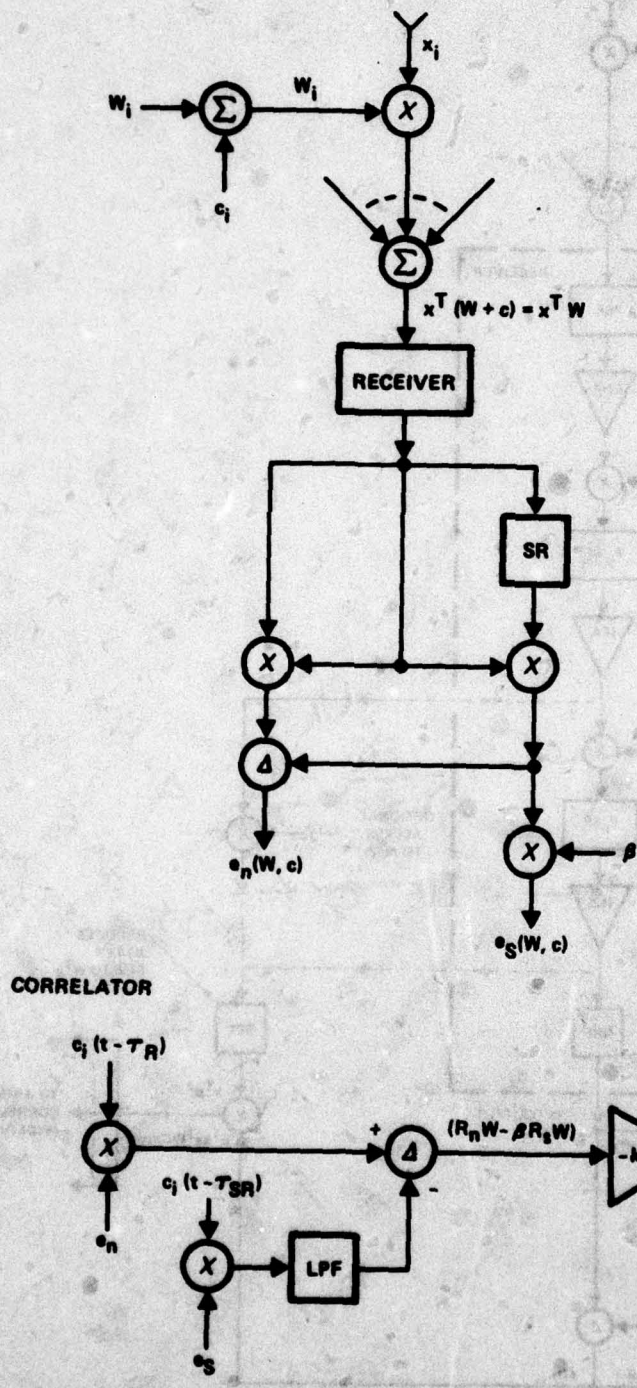


Figure 3.3-1. Perturbational PSF Algorithm

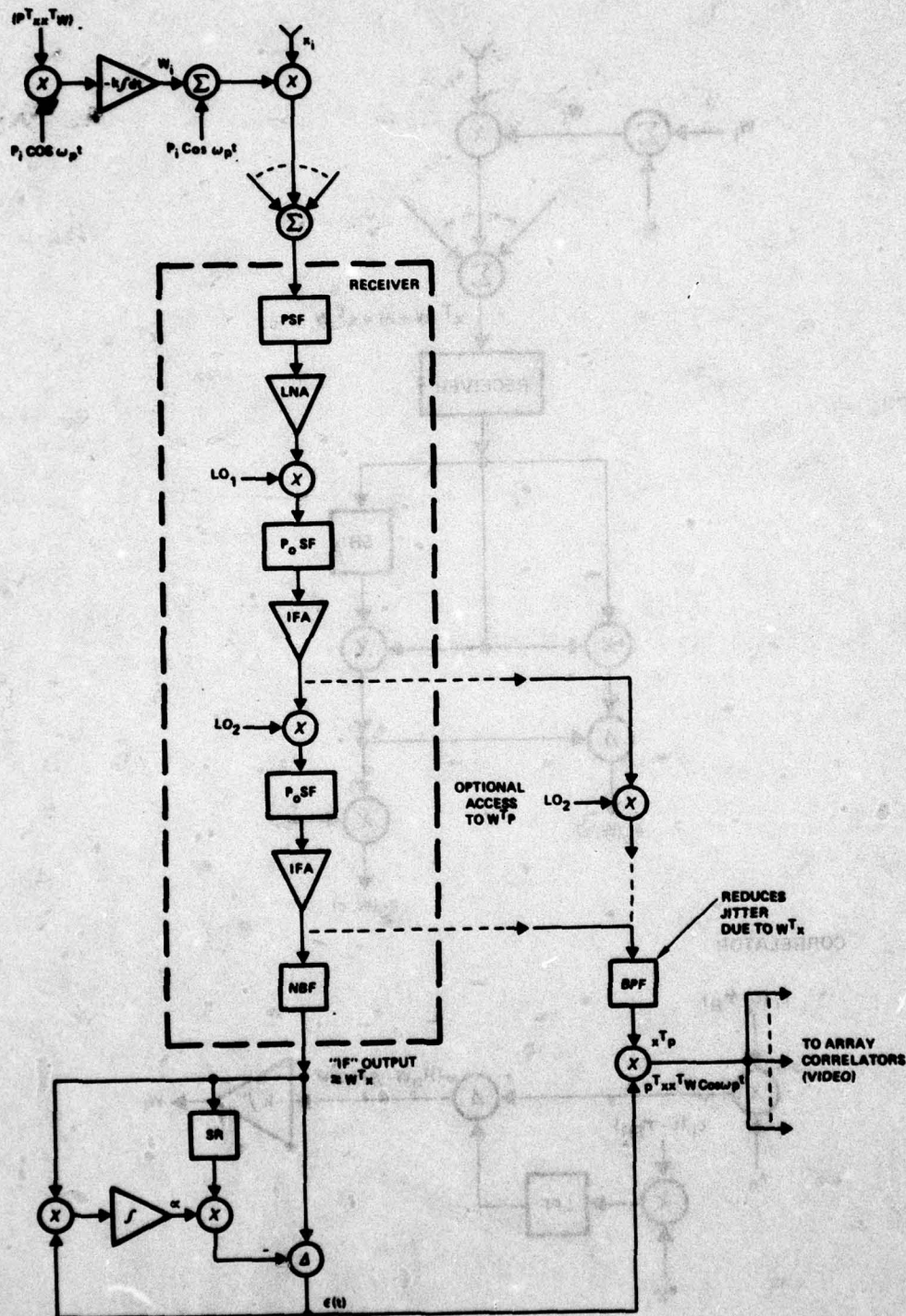


Figure 3.3-2. PSF Algorithm Using FDMA Perturbational Sequences

element. Of course, integration of this gradient provides the desired weight value.

Since the perturbational terms can be biphase orthogonal sequences, the "array correlators" can be simply digital sign changers. In an all digital system, the integrators are replaced by summers. It may be convenient to achieve the array error formation with a conventional analog signal identifier.

Some advantages of this technique are as follows. No RF correlation is required. A single receiver can be used to amplify the weak RF signals. All necessary frequency conversions need be done only once. The performance measure device can be replaced by a conventional PSF error formation circuit alleviating the time delay and error susceptibility of the performance measure device. Correlation and weight control circuitry can be realized in an all digital form if desirable. Finally, the array output is not subject to weight perturbation errors nor is the weight control circuitry subjected to cross correlation errors with the array output.

3.3.3 Random Search

In these paragraphs, we establish an expected value equivalence between a random search algorithm and normalized gradient algorithm. The block diagram of this processor is shown in Figure 3.3-3. Output of the performance measure device has already been calculated in Section 3.2.1. For convenience, we repeat that equation, (3-48) here. We have

$$R(W,C) = \left(\frac{W^T R_s W}{W^T R_x W} \right) + C^T \nabla_W \left(\frac{W^T R_s W}{W^T R_x W} \right) = Q + C^T \nabla Q. \quad (3-48)$$

Recall that this equation expresses the change in the performance measure due to the application of a perturbation C. We assume that the perturbations are zero mean normally distributed random sequences.

According to the conventional random search procedure, if a trial vector C caused the performance measure to increase, then the

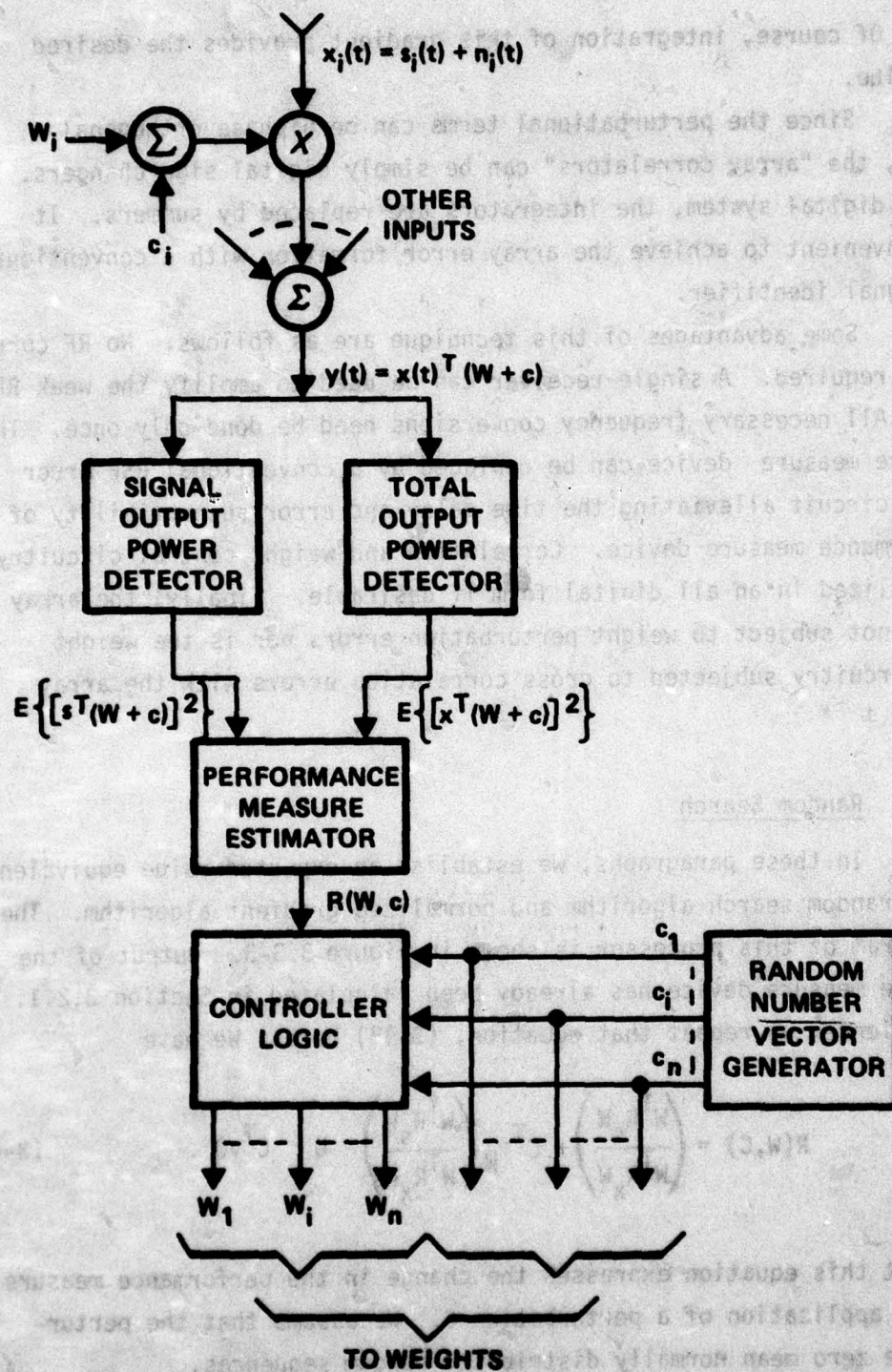


Figure 3.3-3. Random Search Algorithm Conceptual Block Diagram

trial value is added to the weight vector W . Conversely, a trial value C_i which caused the performance measure to decrease is evidently a move in an undesirable direction and therefore C_i is usually subtracted from the current weight vector W (a variation of this procedure is to simply allow W to remain unchanged if no improvement is obtained. We can easily show in the following analysis that this latter procedure is only half as good as the former one). We may express these verbal statements as follows.

$$W_{i+1} = W_i + C_i \quad \text{for } R(W, C) \text{ increased}$$

3-77

$$W_{i+1} = W_i - C_i \quad \text{for } R(W, C) \text{ decreased}$$

Referring to (3-48), it can be seen that any change in the performance measure R for a given value of weight W is due to the second term; the projection of C on the gradient of W . Therefore (3-77) may be written as

$$W_{i+1} = W_i + C_i \left[\text{SIGN}(C_i^T \nabla Q) \right]. \quad 3-78$$

Where the function SIGN is either ± 1 depending upon the algebraic sign of its argument. One can conveniently compute the sign of a variable by dividing that variable by its magnitude. Thus, (3-78) becomes

$$W_{i+1} = W_i + \frac{C_i C_i^T \nabla Q}{|C_i^T \nabla Q|}. \quad 3-79$$

Furthermore, note that the magnitude of a variable can also be obtained by taking only the positive root of that variable squared. Equation (3-79) may be rewritten as follows

$$W_i = \frac{C_i C_i^T \nabla Q}{(\nabla^T Q C_i C_i^T \nabla Q)^{1/2}}. \quad 3-80$$

The change in weight value from the i^{th} to the $i^{\text{th}}+1$ iteration is calculated from (3-79) by subtracting W_i from each side of that equation. Equation (3-80) therefore expresses the change in weight vector at the i^{th} iteration.

At this point, an important question is asked. Is there a deterministic factor in the change of the weight vector? To ask this question in another way let us propose an experiment. Consider that we start at an initial point W_s . Suppose we generate a succession of random vectors C and thus a succession of weight changes W until we arrive (if ever) at a specified value of the performance measure R . Then, let us suppose that the experiment is repeated a large number of times. In these repeated trials, would one be assured of achieving the same performance measure goal? Additionally, would this multitude of paths have some deterministic path in common?

It is important to recognize that we may express the experiment alternatively as follows. Suppose we fix the value of W and apply a succession of random vector perturbations, C , again calculating the multitude of weight vector changes ΔW but not allowing these changes to occur; that is, the algorithm is not permitted to adapt. Again a question is asked; Is there an expected value of weight vector change for this particular weight value? If so, one can evaluate expected weight changes for all possible values of the weight vector W and for any arbitrary sequences of the random vector C . When completed, this procedure will yield results equivalent to the first postulated experiment.

The second postulated experiment may be conducted mathematically by taking the expected value of ΔW . We have

$$E(\Delta W) = E \left[\frac{C(t)C^T(t)\nabla Q}{(\nabla^T Q C(t)C^T(t)\nabla Q)^{1/2}} \right] = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \frac{C(t)C^T(t)\nabla Q dt}{(\nabla^T Q C(t)C^T(t)\nabla Q)^{1/2}} \quad 3-81$$

For Fixed W

For brevity, we will write the gradient of Q as the vector V. That is;

$$V = \nabla_W(Q). \quad 3-82$$

Furthermore, note that (3-81) can be written as follows:

$$E(\Delta W) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \nabla_V (V^T C(t) C^T(t) V)^{\frac{1}{2}} dt. \quad 3-83$$

Where we have used the fact that

$$\nabla_V (V^T C C^T V)^{\frac{1}{2}} = \frac{C C^T V}{(V^T C C^T V)^{\frac{1}{2}}} \quad 3-84$$

We note that the gradient of the term in parentheses with respect to the vector V does not have any particular physical significance. We are using this representation as a mathematical tool.

Since W is held fixed in (3-83), only the perturbation C is a function of time. Consequently, the gradient operation and integral operation may be exchanged in order. We get

$$E(\Delta W) = \nabla_V \lim_{T \rightarrow \infty} \int_0^T (V^T C(t) C^T(t) V)^{\frac{1}{2}} dt. \quad 3-85$$

One may note that the term in parentheses in (3-85) may be written as a magnitude. One has

$$E(\Delta W) = \nabla_V \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T |V^T C(t)| dt. \quad 3-86$$

We expand the term $V^T C(t)$ and get

$$V^T C(t) = V_1 c_1(t) + V_2 c_2(t) \dots V_n c_n(t) . \quad 3-87$$

Note that the V_i are components of the gradient Q and are fixed by W . Furthermore, according to our original problem description, the c_i are normally distributed random variables. It is well known that the sum of normally distributed random variables is itself a normally distributed random variable. Furthermore, the expected value of such normally distributed random variables may be obtained from an integration of a probability distribution function instead of an integration over time. The integral of Equation (3-86) may be written as follows:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T |x| dt = \frac{1}{\sqrt{2\pi} \sigma} \int_{-\infty}^{\infty} e^{-\left\{ \frac{x^2}{2\sigma^2} \right\}} |x| dx . \quad 3-88$$

Where we have let x represent the term $V^T C(t)$.

The integral on the right hand side of (3-88) may be computed by separately evaluating its argument for negative and positive x . We get

$$\begin{aligned} \int_{-\infty}^{\infty} e^{-\left\{ \frac{x^2}{2\sigma^2} \right\}} |x| dx &= \int_{-\infty}^0 e^{-\left\{ \frac{(-x)^2}{2\sigma^2} \right\}} (-x) dx + \int_0^{\infty} e^{-\left\{ \frac{(x)^2}{2\sigma^2} \right\}} (+x) dx \\ &= 2 \int_0^{\infty} e^{-\left\{ \frac{x^2}{2\sigma^2} \right\}} x dx \end{aligned} \quad 3-89$$

where we have used the fact that $(-x) = |x|$ when $x < 0$.

This important result allows one to dispense with the magnitude operator on x . The right hand side of (3-89) is easily integrated and substituted into (3-88) yielding

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T |x| dt = \sqrt{\frac{2}{\pi}} \sigma \quad 3-90$$

The parameter σ is, naturally, the standard deviation. We must now calculate this parameter in order that Equation (3-90) be useful. There are several ways to proceed, but the simplest is apparently through recognition of the fact that the square of the standard deviation is the average power (i.e., mean square value) of the function. Thus, one may write simply that

$$\sigma^2 = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T |x|^2 dt = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T (V^T C C^T V) dt \quad 3-91$$

Since V is static, the V terms in (3-90) may be moved outside of the integral. One gets

$$\sigma^2 = \lim_{T \rightarrow \infty} \frac{V^T}{T} \int_0^T [C(t) C^T(t)] dt V \quad 3-92$$

Let us examine the integral of the matrix $C C^T$ in detail.

One has

$$\sigma^2 = \lim_{T \rightarrow \infty} \frac{V^T}{T} \int_0^T \begin{bmatrix} c_1^2(t) & c_1(t)c_2(t) & \dots & c_1(t)c_n(t) \\ c_1(t)c_2(t) & c_2^2(t) & & c_2(t)c_n(t) \\ \vdots & & \ddots & \vdots \\ c_1(t)c_n(t) & \dots & & c_n^2(t) \end{bmatrix} dt V \quad 3-93$$

Since the individual sequences are independent normally distributed random variables and are thus uncorrelated with each other, the expected value of the off-diagonal terms is zero. The diagonal terms are, of course, the expected value of a given sequence squared i.e., its average power. We assume that by design the power of all perturbational sequences is equal to a value k^2 . Consequently, the expected value of the matrix in (3-93) reduces to k^2 times the identity matrix. One gets

$$\sigma^2 = V^T(K^2 I)V = k^2(V^T V) . \quad 3-94$$

One may now return to the immediate calculation of the expected value of the weight change ΔW . We use the value of σ^2 from (3-94) and the results of integration from (3-90) in (3-85) yielding

$$E(\Delta W) = \nabla_V \left\{ \sqrt{\frac{2}{\pi}} k (V^T V)^{\frac{1}{2}} \right\} . \quad 3-95$$

Finally, the gradient operation indicated in (3-95) is performed yielding the expected value of the weight change ΔW . One gets

$$E(\Delta W) = \sqrt{\frac{2}{\pi}} \frac{kV}{(V^T V)^{\frac{1}{2}}} = k \sqrt{\frac{2}{\pi}} \frac{\nabla \left(\frac{S}{S+N} \right)}{\left(\nabla^T \left(\frac{S}{S+N} \right) \nabla \left(\frac{S}{S+N} \right) \right)^{\frac{1}{2}}} . \quad 3-96$$

Where we have used the definition of V given in (3-82) to obtain the right hand side of (3-96).

The very interesting result expressed by Equation (3-96) is that the expected value of weight change at a given point on the error surface is directly proportional to the normalized gradient of that surface. The inescapable conclusion is that random search is not random at all. This result has further importance in that one may now compute expected value transients, determine conditions for algorithm stability,

calculate weight jitter and weight misadjustment penalties and furthermore, determine whether or not the algorithm converges to optimum solutions. All of these calculations are facilitated by virtue of the fact that the expected value of the weight change is entirely deterministic.

We also note from observation of (3-96) that the rate of adaptation is controlled by the amplitude of the RS parameter, k . Specifically, the factor is $k\sqrt{\frac{2}{\pi}}$.

Earlier in this derivation, it was remarked that a search procedure which adds a random trial C_i to the current weight vector if the performance measure is improved and does nothing otherwise is only half as efficient as one which subtracts unsuccessful trial perturbations. Observe that we can represent this "add if improved" algorithm mathematically by adding 1 to the sign function and dividing the result by two. If the sign is positive, the coefficient of C_i is +1. Alternatively, if sign is negative, the coefficient is zero.

$$\Delta W_i = C_i \left[\frac{1 + \text{SIGN}(C_i^T \nabla Q)}{2} \right] = \frac{C_i}{2} + \frac{C_i C_i^T \nabla Q}{2(\nabla^T Q C_i C_i^T \nabla Q)^{\frac{1}{2}}} \quad 3-97$$

When expected values are taken of Equation (3-97), it is readily seen that the term $\frac{1}{2}C_i$ vanishes. Additionally, the second term is seen to be exactly that of the earlier algorithm, Equation (3-80), except for the factor of $\frac{1}{2}$. Consequently, the procedure represented by (3-97) progresses only half as fast toward the final solution as that expressed by (3-80).

3.3.4 Log Power Algorithm

The utility of the log power technique is primarily dependent upon the availability of automatic gain control amplifiers having logarithmic transfer functions. Since such amplifiers are commonly available for the range of frequencies considered here, it was decided to design a procedure to capitalize on this availability and thus avoid the multiplications

implicit in the PSF algorithm by doing additions instead. Such an exchange of additions for multiplications was especially desirable for the microprocessor simulation utilizing the 8080 due to the lack of an internal multiply command.

A block diagram of the log approach is given in Figure 3.3-4. The adaptive array weights and perturbational inputs are not shown. It is assumed that the array output enters at the top of the figure and is conveyed to a total power detecting multiplier and to a signal recognizer to facilitate detection of the desired signal power. This circuit was described in detail in Section 3.3.1. The output of these power detectors must be filtered so as to avoid the possibility of negative inputs to the log devices. These outputs could be taken from AGC amplifier control voltages, with the signal power AGC control being fed from a signal recognizer.

In principal the iterative equation for a single weight W_i is formed as follows:

$$\Delta W_i = -K \left\{ \left[\ln(P_x + \Delta P_x) - \ln(P_x) \right] - \left[\ln(P_s + \Delta P_s) - \ln(P_s) \right] \right\} \quad 3-98$$

where ΔP is either the power change due to the "three level" orthogonal sequence described in 3.2.2.3.1 or a decorrelated general perturbational term. We assume a perturbation amplitude of δ . Grouping the log terms and simplifying yields

$$\Delta W_i = -K \left[\ln \left(1 + \frac{\Delta P_x}{P_x} \right) - \ln \left(1 + \frac{\Delta P_s}{P_s} \right) \right]. \quad 3-99$$

If the power changes due to the perturbation are small compared with the power output terms due to the nominal weight values, then the log can be approximated using the following relationship.

$$\ln(1+\epsilon) \approx \epsilon. \quad 3-100$$

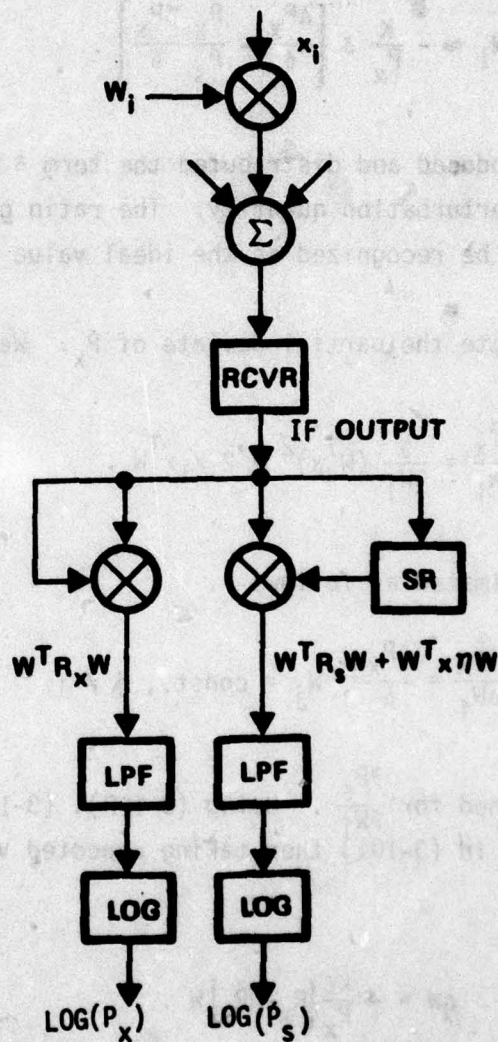


Figure 3.3-4. Log Power Algorithm Block Diagram

Applying the approximation to (3-99) gives

$$\Delta W_i \approx - \frac{K}{P_x} \delta \left[\frac{\Delta P_x}{\delta} - \frac{P_x}{P_s} \frac{\Delta P_s}{\delta} \right]. \quad 3-101$$

Observe that we have introduced and distributed the term δ which is the amplitude of the weight perturbation quantity. The ratio of the static power terms P_x and P_s can be recognized as the ideal value of the PSF parameter alpha.

Let us calculate the partial deviate of P_x . We get

$$\frac{\partial P_x}{\partial W_i} = \frac{\partial}{\partial W_i} (W^T x)^2 = 2 x_i x^T W. \quad 3-102$$

The partial can be approximated as follows

$$\frac{\partial P_x}{\partial W_i} \approx \frac{\Delta P_x}{\Delta W_i} = \frac{\Delta P_x}{\delta}, \quad W_j = \text{const.}, \quad j \neq i \quad 3-103$$

A similar result is obtained for $\frac{\partial P_s}{\partial W_i}$. Using (3-102), (3-103) and the appropriate terms for ΔP_s in (3-101) then taking expected values leads to the desired algorithm

$$\Delta W = - \frac{K\delta}{P_x} [R_x - \alpha R_s] W. \quad 3-104$$

While this approach was used with some degree of success in the microprocessor simulation, it has potential drawbacks in practice because it will almost certainly be sensitive to errors in the logarithmic amplifiers, particularly differential tracking errors. Additionally, the presence of a lowpass filter will certainly slow the algorithm. In retrospect, such slowing is undesirable and unnecessary in view of Section 3.3.1 results. Nevertheless, this technique should be kept in

mind as a possible approach in other applications where it is important to replace multiplication and division by addition and subtraction.

This chapter is devoted to the presentation of a hypothetical system design which is suitable either for upgrading presently unprotected communications networks or for new design of advanced systems. Its basic anti-jam capability is founded in adaptive null steering, but arbitrary degrees of additional spread spectrum AJ can be added as desired. An important advantage of this approach is that it uses existing communications radio sets which need only minimal modification. The adaptive processor controller is simply and inexpensively realized in digital microprocessor circuits. A novel low loss RF weighting technique based on utilization of pairs of phase shifters may allow the elimination of RF amplifiers

We begin this chapter with a short discussion of the postulated system problem in Section 4.1. Then, the selected approach to the problem follows immediately in Section 4.2. The bulk of this section follows in 4.3, where a detailed description is given of the user level communication set/adaptive processor design. Topics discussed here include the array antenna configuration and RF weight design in 4.3.1 and modifications required by the radio set in 4.3.2. The adaptive array processor is presented in substantial detail in Section 4.3.3, where we discuss the algorithm to be used, perturbational sequence design, sampling of the receiver IF output, design of the signal recognizer, interface of the signal recognizer with the modem, utilization of the radio set automatic gain control, and finally, organization of the microprocessor controller, including a detailing of the major integrated circuits required. Brief comments regarding the facilitation of modem synchronization are made in 4.3.4, and then to conclude this chapter, a review of the several possible a priori discriminant functions which were considered is given in 4.3.5, including a justification of the selected multiplicative discriminant technique.

4.1 Postulated System Problem

The intent of this section is to provide as general a system level problem statement as possible while satisfying an original goal of this study; namely, to provide null steering AJ capability for a previously unprotected narrowband radio link. The following is a brief statement of the major guidelines used in this design effort.

4.1.1 Low Cost/High Quality Null Steering Anti-Jam

An adaptive null steering system will provide the basis for anti-jam capability. This is in contrast to most contemporary systems which are principally dependent upon a spread spectrum signalling technique for their fundamental AJ capability.

It is doubtlessly true that one can design high-quality adaptive array systems if cost is not a primary factor. An important goal of this study was to find methods of building adaptive array AJ systems which sacrifice little if any null depth and signal maximization quality yet which are potentially orders of magnitude less expensive than conventional designs.

4.1.2 Complementary Spread Spectrum Anti-Jam Capability

The system design is to allow for arbitrary bandwidth spreading techniques. The design is to be such that the decibel improvement due to spectrum spreading compliments the decibel improvement due to the null steering. Depending upon the type of communication network (for example, AM voice vs. digital data), the spectrum spreading may or may not reduce the probability of message intercept. The design presented here does not preclude the transmission of secure messages but its primary emphasis is to enhance a system's anti-jam capability.

This topic can be clarified through use of a mathematical example. Suppose spectrum spreading is to be introduced into an AM voice communications network. The conventional AM signal may be spread as follows:

$$S(t) = AC(t)[1+M(t)]\cos\omega_c t.$$

4-1

Inspection of the above equation reveals that the bandwidth required for this signal can be determined principally by the term $C(t)$ and that the usual despreading processing gain can be achieved at the receiver by multiplication by the same synchronized function $C(t)$. Note, however, that the envelope of this waveform continues to contain the message information; furthermore, it can be detected although with reduced amplitude by a conventional receiver. Therefore, little security against intercept is provided by this technique.

Alternatively, an FM voice or a biphasic digital data link (perhaps FSK) would have its modulation information distributed throughout the spread spectrum and therefore would be much more difficult to intercept.

4.1.3 Existing Communication Sets

There presently exists a large number of high quality communication sets which have no inherent AJ capability. Most state-of-the-art adaptive array systems in effect provide relatively high quality radio receivers in each of the inputs to be weighted. This obviously is a very expensive approach. Not only is the conventional adaptive processor expensive, it obsoletes the receiver it replaces. Furthermore, it is evident that if an existing receiver can be effectively utilized in realizing the adaptive algorithm, then the cost of a large number of expensive RF and IF components is saved in addition to size, weight and power.

Consequently, the AJ null steering subsystem is to be designed so as to utilize a radio receiver as an inherent component. This approach makes the design suitable for either upgrading existing communications networks or for use in a new system design.

4.1.4 Compatibility

Although it is not necessary, it may be desirable in some instances, particularly narrowband HF, VHF and UHF radio communication networks, to provide null steering AJ capability in a progressive manner. Some users will have an immediate need for an AJ capability, yet other lower priority users might only occasionally if ever, need an AJ capability.

Three levels of AJ protection are to be provided. The first level, having zero AJ, would encompass those users which do not have the adaptive antenna features of the system. Nevertheless, these users are enabled to receive the same signal that protected users receive. The second level of protection is provided strictly by the adaptive null steering system. Those users having modified communication sets and the adaptive array would receive this protection. A third class of user would receive highly spread signals that would ordinarily not be receivable by the lower two classes of system user. This most protected user would receive the combined benefits of null steering AJ and spread spectrum techniques.

It is emphasized that this compatibility requirement is not a necessary feature of the system design. It is, instead, a capability that can be provided if desired.

4.1.5 Present System Features

This requirement is directly aimed at satisfying the needs of typical HF, VHF and UHF communication networks. The two principal features to be retained are first, a conferencing capability and second, a frequency-division multiplex communications network (that is, conventional frequency assignments).

In conferencing, several users may be transmitting simultaneously on a given frequency. It is desirable to provide for simultaneous reception of these signals. When spread spectrum techniques are used in such a communications network, it is much more difficult to ensure simultaneous reception of these signals with a single channel radio.

As will be seen, however, this feature can be provided for at the cost of a modest S/N penalty.

In a conventional unprotected communications network, nearby users on different frequencies do not interfere with each other because their transmissions do not substantially overlap in frequency. When spectrum spreading is used, previously acceptable frequency separations may no longer be adequate due to the additional bandwidth requirement of the spread signal. Under these conditions, nearby users tend to produce an interference signal which overwhelms the reception of weak distant desired signals. If possible, we would like this system design to alleviate this near-far problem without drastic change of existing frequency usage patterns.

4.2 Selected Approach

In this section we will describe on the system level an approach which is capable of meeting all of the goals just listed. To a large extent, only modest penalties are incurred through encompassment of all of these features. In the discussions that follow, both here and in the detailed design section (4.3), the penalties associated with the particular choices will be identified and evaluated. System level considerations pertaining to the adaptive processor are treated first in Section 4.2.1. Then, the selected multiplicative discriminant function and its substantial impact on system performance is covered in 4.2.2. Discriminant function tradeoffs are not covered here; instead, the reader is referred to 4.3.5 for an in-depth review of alternative discriminant functions from which the selected approach was taken.

4.2.1 Selected Algorithm

In keeping with the requirement of retaining the existing radio set and using it as an integral part of the adaptive processing subsystem, we have selected a perturbational type adaptive algorithm. Specifically, the multiple simultaneous signal PSF algorithm is the one of choice. A complete theoretical treatment of this procedure is found in Chapter 3.0.

Salient reasons for the choice of this algorithm are: First, it is capable of obtaining maximum S/N ratio for arbitrary signal bandwidths. Furthermore, it is easily adapted to the maximization of several simultaneous desired signals as is required for conferencing. Third, it is easy to normalize so that in the absence of a desired signal, it reverts to a suppression algorithm. (Other algorithms such as the LMS turn off all weights under the zero signal condition.) Lastly, the adaptation transient of this algorithm shows faster response to the desired signal in the early stages of adaptation, a property not shared by the LMS algorithm.

Due to the fact that this is a perturbational algorithm and the gradient information necessary for its adaptation must pass through the radio receiver's bandpass filters, it is evident that this is a serial-type processor. In contrast, when RF or IF correlation is used, gradient information is extracted in a parallel fashion. We, therefore, expect an adaptation time penalty with respect to this "ideal" approach. This topic is carried further in Section 4.3.3. Additionally, the weight perturbations necessary to extract gradient information cause a degradation in S/N ratio from the optimum obtainable. In this application, such degradation can be made negligibly small. We present simulation data showing this effect in Chapter 5.0.

4.2.2 Selected Discriminant Function

In order that the adaptive processor be able to unambiguously identify the desired signal or signals, it is necessary for one to transmit a unique discriminant function. We have selected a biphase pseudo-random multiplicative function which we will refer to as $C(t)$. Thus, if the existing system waveform is $q(t)$, then the new system waveform is

$$S(t) = C(t)q(t) .$$

4-2

The signal $q(t)$ is allowed to be ordinary AM, FM, FSK, etc. It is evident that if $q(t)$ is an AM signal, then such spreading does not

substantially reduce the probability that the signal can be intercepted. Alternatively, if $q(t)$ is a biphase digital signal, conventional security measures can be taken.

Application of the function $C(t)$ in this manner allows great flexibility in the hardware methods which can be utilized, thus the upgrading of an existing system is facilitated. For example, if the new system waveform is too widely spread in frequency for reception by the existing receiver, then some degree of despreading will be required. If despreading is accomplished prior to introduction of the signal to the communications set receiver, it can be accomplished with a low loss ferrite phase shifter, switched line phase shifter or a mixer. Alternatively, despreading can be accomplished within the receiver by phase shifting a LO line. In some cases, despreading is not required at all. This topic is discussed shortly.

The discriminant function $C(t)$ will have either of two forms depending upon whether the system accommodates the third level of user (a user obtaining spread spectrum AJ processing gain as well as null steering AJ). For the compatible system having only null steering protected users or unprotected users, the discriminant function is a simple pseudo-random sequence having about the same bandwidth as the desired signal before the multiplication operation. We will designate this discriminant as the slow sequence $C_s(t)$.

If the system is to accommodate the third class user, then the discriminant function $C(t)$ has additional structure. We will specify the following:

$$C(t) = C_s(t)C_f(t) \quad 4-3$$

where the term $C_f(t)$ is a rapidly varying pseudo-random sequence having the bandwidth required to realize the desired spread spectrum processing gain.

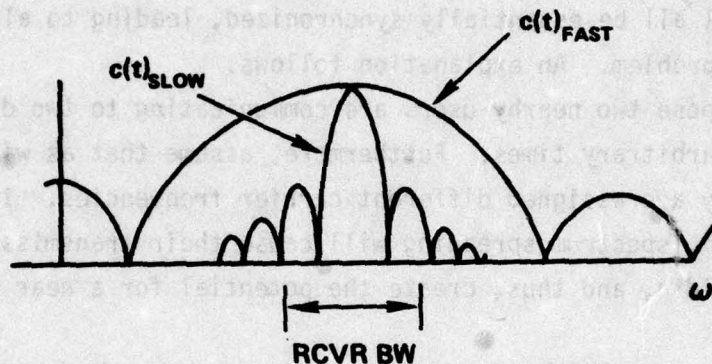
The advantage of having $C(t)$ the product of a slow sequence and a fast sequence is multifold. First, the fast sequence provides the

necessary bandwidth increase for spread spectrum AJ. Second, removal of the fast sequence either between the antenna and the receiver, or in the receiver front end, enables the communications receiver to serve as though it were a modem's narrowband filter. Third, since the slow sequence remains present, it serves as a tag to identify the desired signal to the signal recognizer and the signal-to-noise ratio maximizing algorithm. Fourth, if all system users share the same fast sequence, then nearby users will all be essentially synchronized, leading to alleviation of the near far problem. An explanation follows.

Suppose two nearby users are communicating to two different remote users at arbitrary times. Furthermore, assume that as with present system usage they are assigned different carrier frequencies. In the postulated system, spectrum spreading will cause their transmissions to overlap in bandwidth, and thus, create the potential for a near far problem.

However, if all system users share the same fast sequence, then nearby users synchronized to the fast sequence are essentially synchronized to each other. Thus, transmissions from nearby users are automatically despread into the original FDMA frequency assignments and the near-far problem is substantially alleviated. Naturally, users that are further apart and not communicating with each other will not obtain this despreading advantage. On the other hand, signal attenuation due to range automatically means that the "near-far" problem is not as severe for such users.

A frequency domain illustration of C_f and C_s is found in Figure 4.2-1. Note that the slow term is chosen to have most of its energy within the assumed bandwidth of the communication set receiver. In system operation, the fast spreading term $C_f(t)$ will be removed either at the input to the communication set receiver either through use of a mixer or a phase shifter between the antenna and the receiver or by modification of the receiver's LO. Therefore, only the slow code portion of $C(t)$ will appear at the receiver's intermediate frequency output. This slow discriminant function provides a unique desired signal identifier which,



$$c(t) = c(t)_{\text{SLOW}} \cdot c(t)_{\text{FAST}}$$

- **FAST CODE ENHANCES SECURITY**
- **SLOW CODE FACILITATES SIGNAL RECOGNITION
USING A NARROWBAND RECEIVER**

Figure 4.2-1. Suggested Direct Spread Sequence Structure

when used with a signal recognizer provides sufficient information to enable control of the antenna array by a S/N maximizing algorithm.

In a general application, the path between the receiver's final IF stage and detector would be broken and a despreading mixer or phase shifter inserted to remove $C_s(t)$. (Perhaps with a nominal amplifier gain of 2 or 3 dB). In the case of narrowband AM voice, however, many communication set receivers employ envelope detection. Since the original amplitude modulated signal was phase shifted and not amplitude modulated by $C_s(t)$, the discriminant function is orthogonal to the information as seen by an envelope detector; thus no receiver modification would be required. More sophisticated AM radio sets employing phase lock loop detectors (coherent detectors) would require the despreading operation.

Whenever one employs a spread spectrum signalling technique, a modem is required. Therefore, the receiver's IF output would also be connected to an external modem. The task of this modem would be to generate both the fast and slow pseudo random sequences so as to enable the despreading operation. Here, the communications receiver also serves as a relatively narrowband desired signal filter following the first despreading operation. Thus, substantial modem hardware may be saved by this configuration.

Having the receiver serve as the modem's narrowband filter means that a special technique must be applied to enable the modem to synchronize to $C_f(t)$. No problem is encountered with $C_s(t)$ since this sequence by design has a narrow bandwidth such that, when combined with receiver and transmitter LO uncertainties and possibly doppler frequency shifts, it is well within the receiver passband. Therefore, the receiver output will contain $C_s(t)$, making this term directly available to the modem's acquisition and tracking circuits. Alternatively $C_f(t)$ is removed prior to the receiver filtering, thus the modem is unable to sense timing errors without additional measurement circuitry preceding the filtering. Such special circuitry is described in 4.3.4.

In a jamming environment where phase repetition techniques are not used, it is possible to eliminate the multiplicative fast PN

sequence without loss of null steering AJ capability. It is noted that a jammer having fast tuning and lock-on capabilities would not be particularly troublesome to this system unless it also endeavored to replicate the instantaneous desired signal phase modulation.

An advantage of eliminating the fast code is that in the case of an AM voice communications network, the unmodified, unprotected radios can also receive the coded transmissions if jamming in their locality is not too severe, thus the system is compatible both with null steering equipped and unequipped users. An additional advantage is obtained in that a much simpler modem can be used. Synchronization to the slow code is much easier than synchronization to the fast code.

As is always the case in an encoded multiple access system, synchronization of multiple users is a problem. In the case of the narrow-band AM voice communication systems, such simultaneous treatment of multiple desired signals is necessary to enable conferencing. The situation is illustrated in Figure 4.2-2 where several users labeled U_i are arbitrarily positioned. Several of the users may be closely located such as U_3 through U_6 . These users may all be essentially in simultaneous synchronization with the discriminant function; on the other hand, U_1 and U_2 , if they are sufficiently separated cannot simultaneously be in synchronization with each other and the cluster of users. Therefore, separate modem tracking loops will be required for each distinctly separate communications user. If synchronization to within one-half chip of the pseudo random spreading function is required, then the allowable separation of users can be calculated if the chip time is known. The following table shows the maximum permissible separation of users for given chip times assuming that an additional modem tracking loop is unavailable.

If the principal need for conferencing exists only among relatively closely spaced users, then a single modem tracking loop and single front-end despreading mixer may be sufficient. This problem is substantially eased if only the slow pseudo random code is utilized. Alternatively, use of a single receiver to conference spread spectrum users having different synchronization times means that a noise penalty will be incurred by a system receiver.

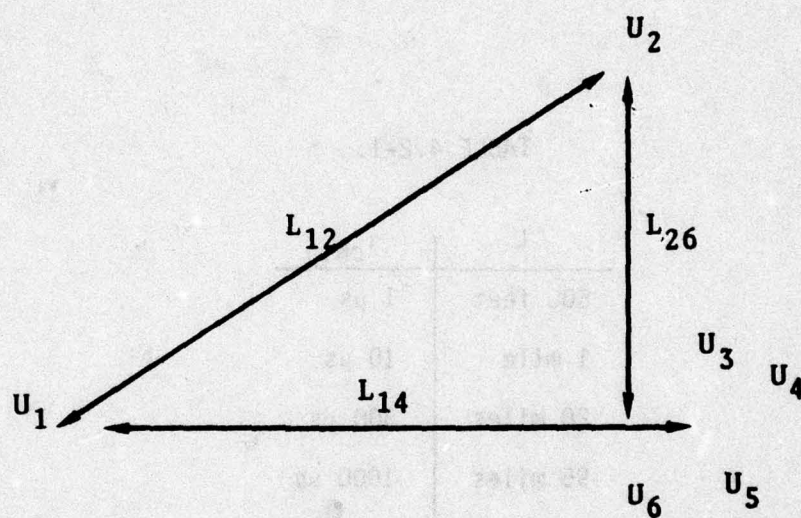


Figure 4.2-2. Separation of Communication System Users

TABLE 4.2-1.

L	τ_{chip}
500 feet	1 μs
1 mile	10 μs
28 miles	300 μs
95 miles	1000 μs

The method by which the simultaneous spread users are conferenced is illustrated in Figure 4.2-3. The single despreading mixer at the communication set receiver's input terminal is supplied by the modem with all of the pseudo random sequences correctly phased respectively for the several desired signals. These sequences are summed and applied to the single mixer. In a manner analogous to the application of summed signals to a limiter, the component of each of the desired signals is reduced by $1/N$ where N is the number of simultaneous desired signals in this operation. Consequently, there is a loss in gain for each individual signal with total noise input remaining unchanged; therefore, noise figure is degraded when more than one desired signal is required. The noise figure penalty for two simultaneous signals is 3 dB while that for four simultaneous signals is 6 dB. In many cases, particularly in a jamming environment, system performance is not determined by thermal noise considerations. Instead it is governed principally by the performance of the AJ system. Consequently, degradation of the receiver's noise figure during conferencing may result in only an insignificant system level performance loss.

4.3 Detailed Design

The following sections describe in detail the critical components needed to assemble the adaptive array AJ subsystem outlined in the preceding section. These details are not totally complete in that specific communications set parameters must be known in order that gain constants, adaptation times, interface devices, etc. can be determined. However, when the engineering data from the computer simulation runs is combined with the design information of this section and parametric details from the communications set to be converted, then the design task can be completed.

Before proceeding with the details, it is helpful to gain an overall "hardware" view of the adaptive array AJ subsystem. A diagram of the major components is given in Figure 4.3-1. At the top of the figure, an antenna array having four elements is connected to a weighting/summing device. We have chosen a paired phase shifter-type

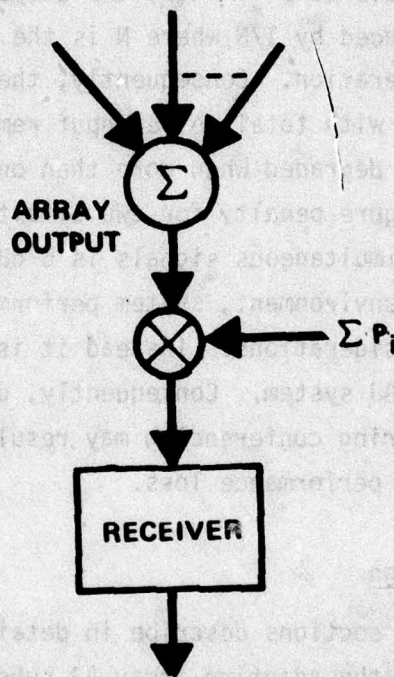


Figure 4.2-3. Despredding of Multiple Simultaneous Desired Signals to Enable Conferencing

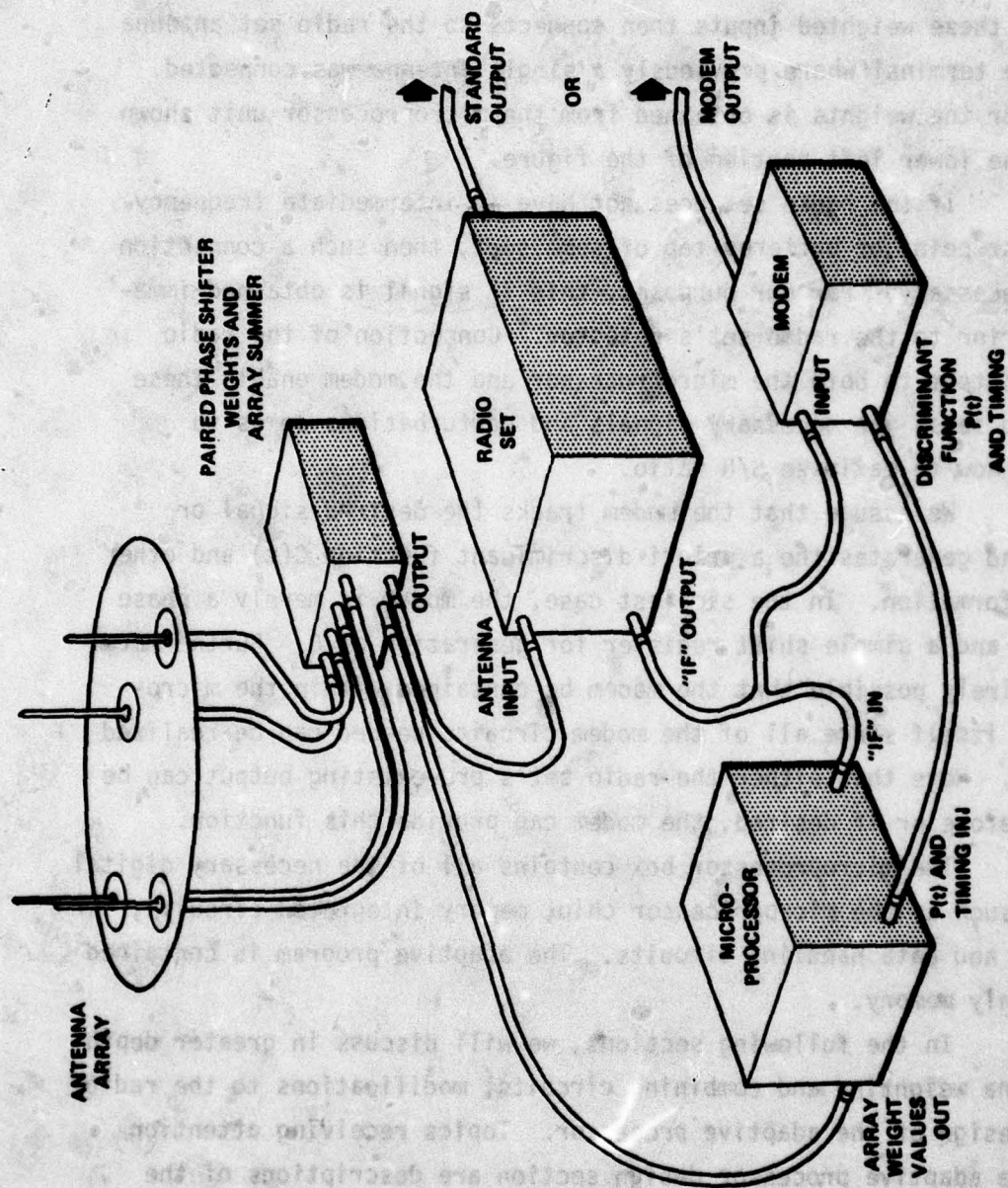


Figure 4.3-1. Pictorial Diagram of A) System Hardware

weight which is potentially low loss and may not require RF amplification. Naturally, it is also possible to use conventional attenuator type weights and RF amplification to make up the gain loss. The summed output of these weighted inputs then connects to the radio set antenna input; the terminal where previously a single antenna was connected. Control for the weights is obtained from the microprocessor unit shown here at the lower left portion of the figure.

If the radio set does not have an intermediate frequency output test point or buffered tap of some sort, then such a connection will be necessary. For our purposes, this IF signal is obtained immediately prior to the radio set's detector. Connection of the radio set's IF output to both the microprocessor and the modem enable these devices to sense the necessary signals and perturbations so as to determine how to maximize S/N ratio.

We assume that the modem tracks the desired signal or signals and generates the a priori discriminant function $C(t)$ and other timing information. In the simplest case, the modem is merely a phase lock loop and a simple shift register for generation of C . Furthermore, it is entirely possible that the modem be contained within the microprocessor itself since all of the modem circuits needed can be realized digitally. Note that either the radio set's pre-existing output can be used as before or if desired, the modem can provide this function.

The microprocessor box contains all of the necessary digital hardware such as the microprocessor chip, memory integrated circuits, buffering and data handling circuits. The adaptive program is contained in read-only memory.

In the following sections, we will discuss in greater depth the antenna weighting and combining circuits, modifications to the radio set and design of the adaptive processor. Topics receiving attention within the adaptive processor design section are descriptions of the algorithm, perturbational sequences used, the correlation discriminant operator function (signal recognizer), the method of sampling the radio set's IF output, the manner in which the sets AGC voltage is used to control the algorithm adaptation rate and design of the microprocessor

circuits. A brief treatment of the modem and a review of discriminant functions receiving consideration concludes this chapter.

In order to facilitate understanding this design on an individual circuit basis as well as to clarify circuit interrelationship, an overall moderately detailed block diagram is given in Figure 4.3-2. The reader is encouraged to refer to this figure as each major circuit is discussed.

4.3.1 Antenna

We use the term "antenna" to mean not only antenna elements but the weighting circuits and array summer as well. In this sense, the antenna is a unit which replaces the pre-existing single antenna to which the radio communications set was connected. Placement of the antenna elements is tailored according to needs of the particular application; for example, some scenarios will benefit from a linear array while others will benefit from a circular configuration. A configuration we favor in many applications has four elements placed approximately 90° apart on a logarithmic spiral. We have found this positioning to be less susceptible to the phenomena of grating nulls due to its lack of periodicity. A symmetrical Y-shaped configuration has been discarded since examination revealed severe grating null effects.

If desired, conventional attenuator-type weights can be employed. If so, it is necessary to provide RF amplification to compensate for the nominal attenuation of these weights. Control of such weights is entirely straightforward and will not receive further attention here.

However, we would like to further discuss a weighting scheme which has promise in this application. We refer to it as a "paired phase shifter" weight. This weighting scheme is capable of generating any complex weight value less than unity. Thus, it is no less general than the attenuator-type weights. Its primary advantage is that it can be set for zero attenuation in contrast to the attenuator type weight which may have as much as 10 dB attenuation in its minimum weight setting condition. In order for either of these weighting schemes to achieve

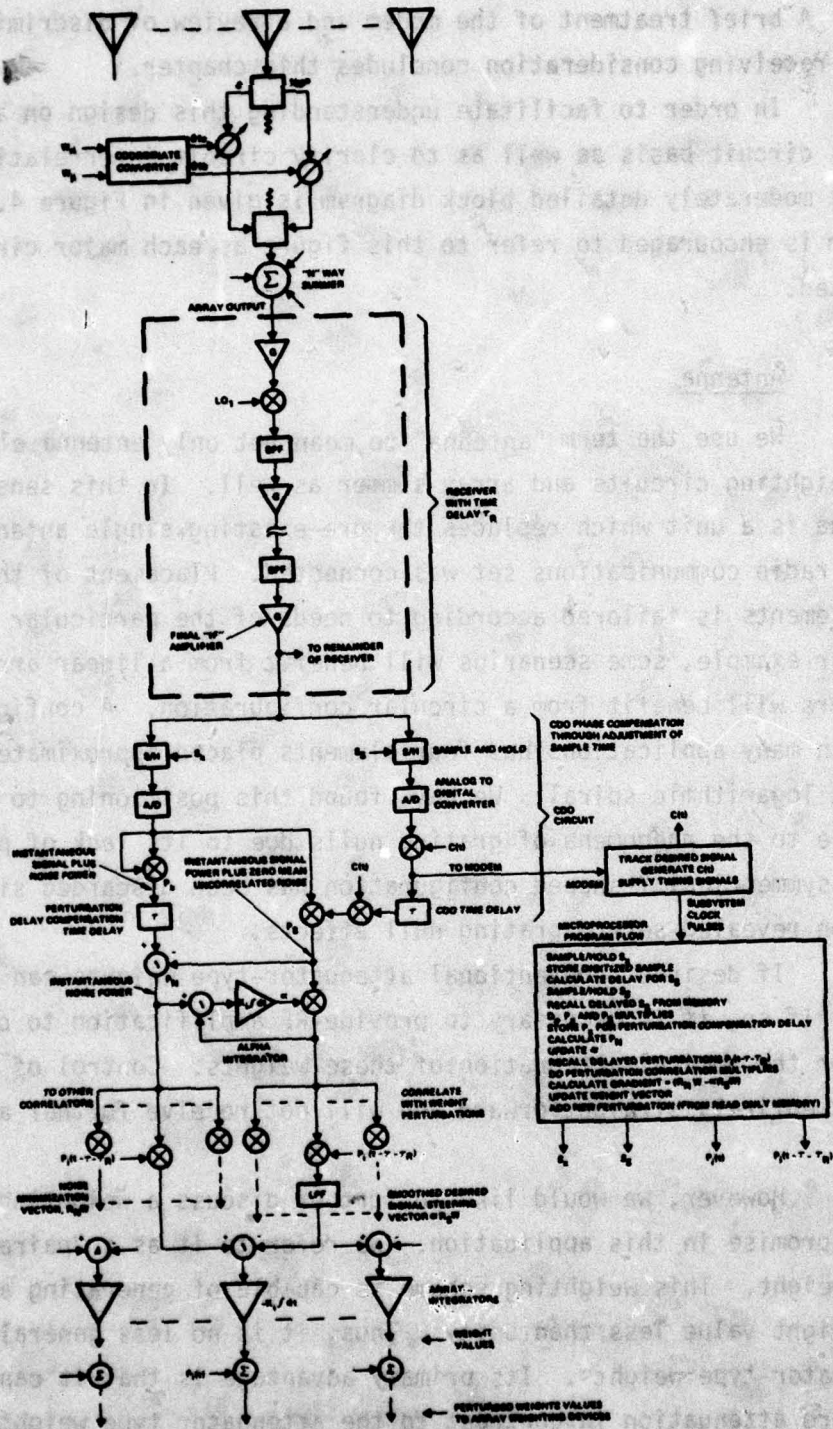


Figure 4.3-2. AJ System Overall Block Diagram

weight values greater than unity, amplification is necessary. Since all of the algorithms we have studied at times require weight values much greater than unity, it is usually necessary that amplification be used with either the conventional attenuator-type weight or with this paired phase shifter weight. However, less amplification is required with the phase shifter device.

Although we have not proven this approach, there is an intriguing possibility that the microcomputer's program can scale the weights as a function of time so that the largest weight is always unity. Then the paired phase shifter weight is capable of meeting the antenna array weighting requirements without the need for amplification. Alternatively, the attenuator-type weights would each require at least a 10 dB gain RF amplifier.

A circuit diagram and a phasor diagram are given in Figure 4.3-3; Part (a) of the figure shows an input to be weighted being fed through a power dividing hybrid. This divided signal is then routed to two different phase shifters ϕ_1 and ϕ_2 . Control lines lead to these phase shifters the voltage of which determines phase shift. Then the phase shifted signals are combined in a second hybrid to provide the weighted output.

The phasor diagram given in Part (b) of the figure illustrates how the paired phase shifter weight works. A point has been selected in the fourth quadrant of the complex plane as the desired weight value. Inspection of the diagram shows that either $\phi_{1a} + \phi_{2a}$ or that $\phi_{1b} + \phi_{2b}$ can achieve this value. The circles are unity amplitude and represent the locus of points achievable by the two phase shifters respectively. One circle is drawn at the origin of the diagram and the second is drawn at the desired weight value. Therefore, the intersections of these two circles determine the pair of phase shifter values which will give the required complex weight. Inspection of this diagram will also reveal that the maximum allowed weight value is obtained when the phase circles just touch. This, of course, is a sum of two but when it is recognized that the input power was divided by two, the input/output

attenuation assuming no loss in the hybrids or phase shifters is 0 dB.

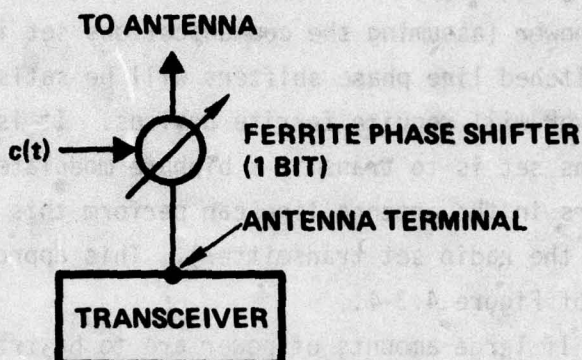
The type of phase shifter to be used is governed by the power handling required. For receive only or transmit/receive given modest transmitted power (assuming the communications set is a transceiver), pin diode switched line phase shifters will be satisfactory. Handling of higher power will require ferrite devices. It is noted that if the communications set is to transmit a biphasic modulated signal, these phase shifters in the antenna line can perform this function (as opposed to modifying the radio set transmitter). This approach is illustrated in part (a) of Figure 4.3-4.

If large amounts of power are to be transmitted and furthermore only one antenna is to be used for transmission (for an omni radiation pattern), then a transmit/receive switch should be included in this circuitry. Such a T/R switch could also facilitate the handling of high transmitter power by switching in a ferrite phase shifter modulator during transmission only.

If the AJ system utilizes a fast pseudo random discriminant function which must be despread before introduction into the communications set, then the phase shifter weights can be phase alternated by the microprocessor to perform this phase demodulation function. However, it is probably much simpler and less demanding of the microprocessor to include a single additional phase shifter in the antenna input line for the communications set. A phase shifter is preferred to a mixer because of its lower loss. It is also noted that the phase demodulation requirements of conferencing can be met by this same inline phase shifter.

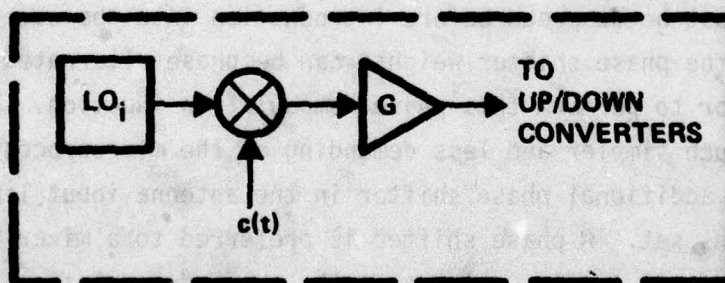
4.3.2 Communications Set Modifications

Relatively minor modifications of the radio is required. The principal change is provision for outputting the radio's final intermediate frequency signal. This coherent waveform is sampled by the microprocessor and used in achieving its S/N maximization algorithm. Additionally, if a spread spectrum modem or, at least, code generating modem is to be used, then it too needs to be connected to this final



- NO TRANSMITTER MODIFICATIONS NEEDED
- MAY BE SLOW

(a)



- REQUIRES RADIO MODIFICATION
- PN BW AS WIDE AS TRANSMITTER FILTERS ALLOW

(b)

Figure 4.3-4. Methods for Applying $C(t)$ to an Existing Transceiver

intermediate frequency output. Many radio receivers provide IF test points which automatically satisfy these needs. If such test points are not available, however, it is a simple change to introduce a buffer amplifier.

Although it is not shown in Figure 4.3-2, the microprocessor controller also requires knowledge of the radio set AGC voltage. The reason for this will be evident when the AGC topic is discussed in Section 4.2.3.5, but basically the reason is as follows. As the jammer is nulled, gain of the radio receiver increases so as to retain a more or less constant output level. Unless the algorithm has knowledge that receiver gain has changed, it will interpret this increased output as either a failure to produce a satisfactory null or that its control efforts have gone in the wrong direction. Through adjustment of the algorithm gain, it is possible to keep the product of receiver gain times algorithm gain a constant. Under these conditions, the adaptive algorithm is stable.

We note that once the communications radio set is opened for circuit changes, it is a relatively simple matter to provide for discriminant function spreading and despreading operations. For example, the first or second local oscillator in the radio can be mixer or phase shifter modulated with a minimum of circuit changes. In some instances, it will be necessary only to break a connection and insert the phase modulation device. If an oscillator line which functions during transmit as well as receive is chosen, then the desired conversion to a spread spectrum transceiver will have been made. This method of spreading/despreading is shown in part (b) of Figure 4.3-4.

In the case of AM voice transmission, if the radio set uses an envelope detector, then its internal demodulation and output circuits can continue to be used. In the case of FM or coherent AM detection, either of two possibilities are attractive. First, a slow code despreading operation can be done by either the microprocessor or the modem and this despread signal rerouted to the radio set where the remainder of its detection and output circuits function as before. Alternatively, the modem can perform the demodulation function.

4.3.3 Adaptive Processor

The following paragraphs describe the adaptive algorithm used, the perturbational sequences which enable gradient measurement, the method by which the receiver intermediate frequency output is sampled so as to provide the necessary performance data to the algorithm, detailed utilization of the receiver's AGC voltage and specification of the micro-processor and its related integrated circuits.

4.3.3.1 Algorithm

The positive signal feedback algorithm modified for the simultaneous reception of multiple desired signals is selected. This algorithm is described in detail in Chapter 3.0 of this report. Given specific information about the communications radio set, precise gain values can be determined from the equations of Chapter 3.0 and the simulation data from Chapter 5.0. Reference to Figure 4.3-2 shows major flow paths of the algorithm and interface with other circuits.

4.3.3.2 Perturbational Sequences

As explained in Chapter 3.0, Algorithms, utilization of a single receiver as an error channel amplifier means that special provision must be made to enable gradient measurement for the algorithm. According to the theory of Chapter 3.0 and the simulation results given in Chapter 5.0, we believe that the perturbational method using Walsh functions is the best choice. Some interesting and useful Walsh function details may be found in Chapter 3.0 in the discussion of Perturbational Algorithms. Since Walsh functions can correlate well with sinusoidal functions, it is recommended that a single biphase pseudo random multiplier be applied uniformly to these sequences as they are used. In extraction of the gradient terms, the pseudo random term is automatically squared and hence does not affect the algorithm. Alternatively, the pseudo random-Walsh perturbation of the weights cannot correlate with an input carrier.

Amplitude of the perturbational sequences must be adjusted so that the gradient is calculated with sufficient accuracy but that weight jitter does not substantially degrade output S/N ratio. Simulation data regarding this tradeoff is given in Chapter 5.0. Specific numerical values of the perturbation will be dependent upon the communications set gain.

Depending upon the communications set's bandpass characteristics, specifically, time delay near band edge, it may be desirable to reduce the rate of the perturbational sequences so as to avoid dispersion of the high frequency components of these perturbations. It is emphasized that if one perturbs the weights too rapidly, the receiver bandpass will not allow these changes to be accurately measured and the algorithm will fail to adapt satisfactorily. Reduction of the rate of the perturbational sequences directly affects adaptation time in that gradient measurement requires a completed perturbational sequence. Regardless, if one conservatively chooses only the middle half of the receiver bandpass for accommodating the perturbational sequences, then practically no time delay dispersion results and algorithm adaptation rate is reduced only by a factor of two from the maximum obtainable.

4.3.3.3 IF Output Sampling

Adequate information for algorithm control is obtained by sampling the receiver IF output at at least twice the receiver bandpass frequency value. Note that it is not necessary to sample at a rate twice the IF frequency because the information required by the adaptive algorithm is only the amplitude and phase envelope of the IF carrier. This information changes at a rate governed by the IF filter bandwidth, not the IF frequency. Undersampling of the receiver IF output, with respect to the IF frequency, is equivalent simply to performing a frequency downconversion to near baseband. We show this as follows. Let the sampled waveform $S(t)$ be expressed as in the following equation:

$$S(t) = A(t)\cos\{\omega_{IF}t + \theta(t)\} \delta(\cos\omega_s t) \quad 4-4$$

The terms $A(t)$ and $\theta(t)$ are the amplitude and phase information of interest and are terms which must be sampled at a rate of at least twice the receiver's IF passband. The term ω_{IF} is the IF frequency of the receiver which may be and likely is many times greater than the sampling frequency ω_s . The term $\delta(\cos \omega_s t)$ is a delta function which has a value of unity only when its argument is zero; Otherwise, the delta function is zero. This term thus provides a sampling of the IF output twice per cycle of ω_s . Let us calculate several values of the sampled output. We get

$$\begin{aligned} S(t_0) &= A(t_0) \cos \{ \omega_{IF} t_0 + \theta(t_0) \} \\ S(t_1) &= A(t_1) \cos \{ \omega_{IF} t_1 + \theta(t_1) \} \\ &\vdots \\ S(t_i) &= A(t_i) \cos \{ \omega_{IF} t_i + \theta(t_i) \} \end{aligned} \quad 4-5$$

Where i is the sample number and the time t_i is calculated as follows:

$$\omega_s t_i = \pi i \quad 4-6$$

when (4-6) is substituted into (4-5) we get

$$S(t_i) = A(t_i) \cos \left\{ \omega_{IF} \left(\frac{\pi i}{\omega_s} \right) + \theta(t_i) \right\} \quad 4-7$$

In general, the term $\left(\frac{\omega_{IF}}{\omega_s} \pi \right)$ may be expressed as follows:

$$\frac{\omega_{IF}}{\omega_s} \pi = m(2\pi) + \Delta\omega \quad 4-8$$

where m is an integer and the term $\Delta\omega$ is the quantity by which the frequency ratio times π exceeds an integral number of 2π radians. Using this result in Equation (4-7), yields

$$S(t_i) = A(t_i) \cos\{(\Delta\omega)t_i + \theta(t_i)\} .$$

4-9

It can be seen that the sequence of sampled values is equivalent to having sampled a downconverted intermediate frequency output, the carrier of which would correspond to the smallest difference frequency between an arbitrary harmonic of ω_s and ω_{IF} . Furthermore note that since the amplitude and phase information of interest can change no more rapidly than the receiver bandpass allows, no information useful to the algorithm has been lost through this "undersampling" procedure.

In terms of hardware, we suggest that sampling for the microprocessor be accomplished with a combination sample-and-hold device and analog-to-digital converter. Since very fast sample-and-hold circuits are available, no hardware problem is expected in sampling intermediate frequencies. Furthermore, holding the sampled value allows the use of a slower A/D converter, permitting trades of either accuracy vs. time or cost vs. conversion time.

Although we will treat this topic in greater detail in the following section, LO and transmitter frequency uncertainty necessitates a signal recognizer phase correction. It is evident that control of sampling time is equivalent to introducing a phase shift in the sampled value. Consequently, two sample-and-hold devices are actually used. (Refer to Figure 4.3-2.) The first is used by the algorithm to extract the gradient term $R_x W$ while the second sampling time is delayed with respect to the first so as to introduce the phase correction required for the SR. This leads to the other necessary term $R_s W$.

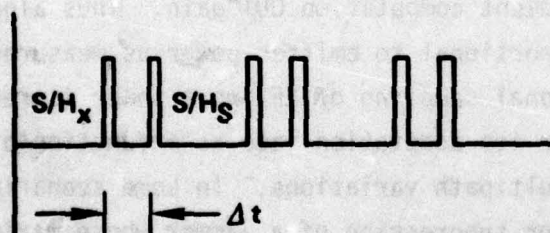
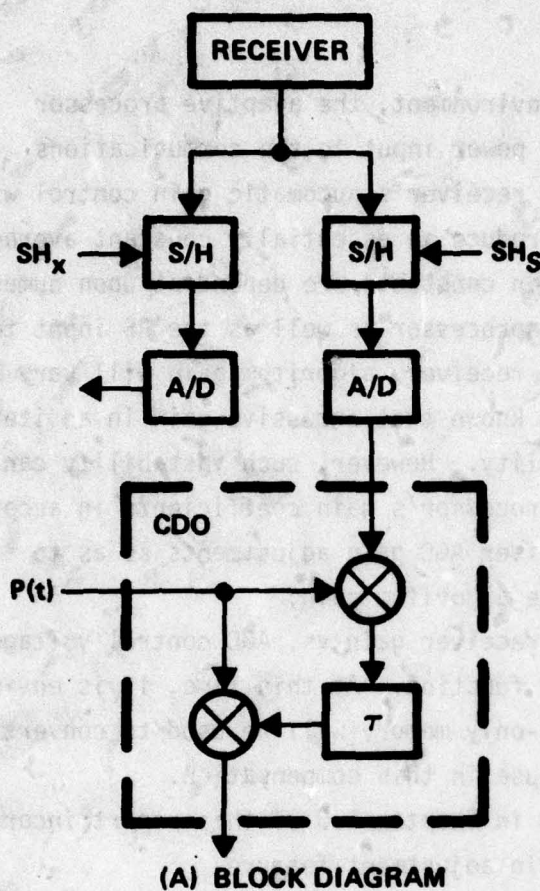
4.3.3.4 SR Realization

Sample-and-hold phasing details are given in Figure 4.3-5, with part (a) of this figure repeating the circuit as shown in 4.3-2. In the signal line, the S/H_s sample-and-hold value after being digitized is multiplied by the discriminant function $C(t)$ and stored (this multiplication is by either +1 or -1 and is realized respectively either by no

change in the digitized value or by complementation of the digitized value). Successive values are also stored in memory until the desired amount of time delay has elapsed according to the microprocessor (or modem) clock. Then, previously stored samples are recalled from memory, multiplied by the current $C(t)$ and used by the algorithm.

In an application such as this where a narrowband SR is utilized, the incoming desired signal carrier frequency uncertainty is such that the term $\omega_s \tau$ may be several times 2π radians. Phase lock tracking loops in the modem (or in the microprocessor) can easily lock to the desired signal carrier even when S/N ratio is too poor to allow communication of information. This loop provides an estimate of the value ω_s . Since τ is known by design, the SR phase term can be calculated. Then, using this information, the SR sample-and-hold is delayed with respect to the primary sample-and-hold so as to introduce the negative of the phase term $\omega_s \tau$; thus the CDO phase term is corrected. This procedure is illustrated in part (b) of Figure 4.3-5.

It is important to note that substantial errors in the estimation of the desired signal carrier frequency can be made without substantial performance degradation of the SR. For example, suppose that the receiver passbandwidth is 25 kHz. Furthermore, let us assume that LO/carrier frequency uncertainty is 10 kHz. If we choose a SR delay of 120 microseconds, the total phase uncertainty from the SR is about 430° . If the modem is not phase locked but instead is merely able to approximately track the desired signal carrier frequency, a misestimation of frequency by 700 Hz causes only a 30° error in the SR phase term. Such an error causes only very small performance degradation of the PSF algorithm. Under more expected conditions, the modem's desired signal carrier tracking loop can be in phase lock even though output S/N ratio in the information bandwidth is too poor to allow information transfer. Given that phase is being tracked, even if with substantial error, the frequency term is essentially correct. Therefore, the SR phase correction term will have negligible error.



WHERE $(\omega_s \Delta t) = (\omega_s \tau) \text{ MODULO } 2\pi$
 AND $\omega_s = \text{DESIRED SIGNAL CARRIER FREQUENCY AT RECEIVER IF OUTPUT}$
 AND $\tau = \text{CDO DELAY}$

(B) TIMING DIAGRAM

Figure 4.3-5. Sample and Hold Circuits for Applying CDO Phase Correction

4.3.3.5 AGC

When, in a jamming environment, the adaptive processor begins to null the interferor, RF power input to the communications receiver will diminish. Then the receiver's automatic gain control will begin to compensate, seeking to produce an essentially constant average power output. Since algorithm gain constants are dependent upon numerical coefficients employed in the microprocessor as well as the RF input to IF output gain of the communications receiver, algorithm gain will vary with the AGC compensation. It is well known that excessive gain in an iterative algorithm will cause instability. However, such instability can be prevented by adjusting the microprocessor's gain coefficients in accordance with the communications receiver AGC gain adjustments so as to obtain a constant overall adaptive algorithm gain.

It is expected that receiver gain vs. AGC control voltage (or voltages) will be a nonlinear function. At this time, it is envisioned that a specifically tailored read-only memory will be used to convert AGC values to actual gain values for use in this compensation.

Simulations reported in Chapter 5.0 of this report incorporate such a constant overall gain adjustment feature.

Once compensated in this manner, the adaptive processor displays a constant RF IN to gradient computation OUT gain. Thus algorithm adaptation speed is linearly proportional to emitter power as measured at the antenna elements. An additional sampling of RF input power is required in order that the algorithm alter its adaptation rate as a function of power changes due to range and multipath variations. In some scenarios, however, the principal need is for suppression of a jammer whose maximum power is moderately well known in advance. In such a situation, it is possible to preset the adaptive array gain in advance so as to ensure adequate jammer nulling. The penalty paid in such an approach is that in the absence of jamming, adaptation rates to a desired signal whose power varies due to range variation and multipath variation is not constant. On the other hand, if this adaptation rate is sufficiently fast to prevent severe desired signal power output changes, then this approach is acceptable.

For the type of problems addressed here, the constant circuit gain approach is desirable; thus we have not provided for a measurement of total antenna element input power.

If such power measurement becomes desirable, two approaches are suggested. First, it is theoretically possible to extract a measurement of $x^T x$ through an appropriate manipulation of the perturbational sequences and perturbed receiver output power. However, for small perturbations (to minimize S/N degradation due to weight jitter) the precision of the $x^T x$ measurement is likely to be poor. The suitability of such computation can be determined by exercising a modified version of the computer simulation program developed for this study.

Alternatively, a power divider, RF amplifier and detector may be connected to an antenna element. This approach is not favored since the power detector would also have to be tuned in order that out-of-band jamming not be measured.

4.3.3.6 Microprocessor

Microprocessor considerations are organized into two parts. A summary of the selected device and related hardware is given first on a general level. It is followed by a more specific summary of how the hardware is applied to realization of the PSF algorithm.

4.3.3.6.1 General Design Features

The hardware proposed to realize the adaptive algorithm is a microprogrammed, bipolar, bit-slice microprocessor using the AMD 2900 family of integrated circuits. This family of chips has established itself as the leader in bit-slice elements and is widely second sourced.

The architecture of the machine is shown in Figure 4.3-6, and is of a general purpose nature with special features added to enhance its digital signal processing capability. As shown in the figure, the processor is built around a central array of four 4-bit AM2901A processing elements. These elements contain 17 words of high speed scratch pad memory, a high speed ALU, and the associated shifting, decoding and

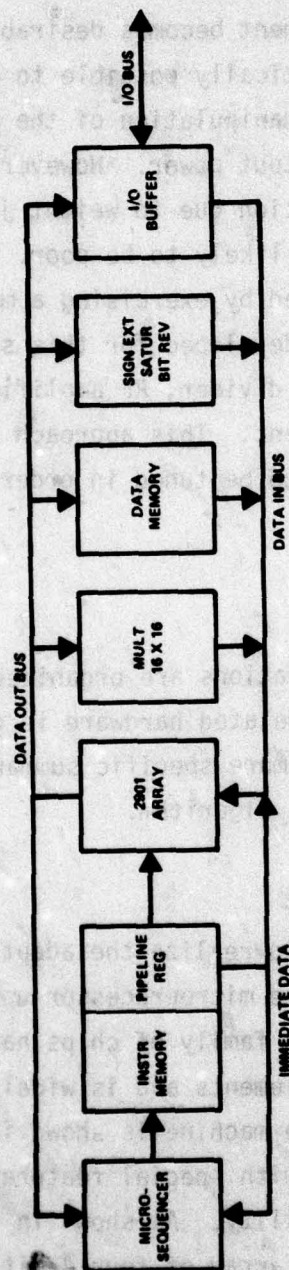


Figure 4.3-6. Selected Microprocessor Architecture

multiplexing circuitry. All movement of data, logical operations, additions/subtractions, and shifting is accomplished within this array.

Emanating from the central element are two unidirectional buses, each 16 bits wide, one for input transfers and one for output transfers. Between these two buses are positioned data memory, a high speed multiplier, bit reverse, sign extend and saturation circuitry. The data memory is fully registered on input and output so that it operates synchronously with the rest of the machine. Most memory transfers are two cycle operations with the address loaded in one cycle and data read or written in a following cycle. A counter is used for the memory address register and provision is included for auto incremented, single cycle memory read or write operations on arrays of data. The high speed multiplier is a single chip TRW-MPY16 unit that performs a full 16x16 bit multiply with a 32 bit rounded product in 200 nsec. The high speed multiplier greatly enhances the digital signal processing capability, for example, a complex multiply can be performed in seven cycles. Another signal processing feature of the machine is program controlled saturation or hard limiting of add/subtract operations. This circuitry incurs no execution time penalty for overflow checks.

The two unidirectional buses in the architecture are also combined to form a single bidirectional bus for communication with external devices. The single bus is used to minimize card pinout requirements.

The microprogram sequencer used in the processor is a single chip unit, the AM 2901. It provides a variety of possible branching, subroutine calls and looping operations. Also included is an internal counter for overhead-free looping operations. This counter/register may be loaded from either the pipeline register or the CPE output. Eight condition codes including interrupt and I/O status sense are provided for condition sequence operations. One level of interrupt is provided via conditional subroutine jumps based on the state of the interrupt line.

The output of the sequencer is the address of the next instruction to be executed. This address is fed to the micromemory or instruction memory. The fetched instruction is loaded into a pipeline register which holds the instruction while it is being executed by the microprocessor. The pipeline register allows faster operation of the

machine in that instruction fetch and instruction execution are performed in parallel, and is completely transparent to the programmer.

4.3.3.6.2 Demonstration Unit Design

A proposed design for a demonstration unit incorporating the adaptive phased array control algorithm is presented in this section. This design is based on the bit slice microprocessor described in previous sections. It includes I/O capability for analog input from the radio receiver and modem and for digital output of the element weights to the phase array antenna. Also included is front panel controls for adjustment of key parameters within the adaptive algorithm.

The physical design of the proposed demonstration unit is shown in Figure 4.3-7. The unit is packaged in a standard 19" rack mount drawer 5½" high. It is completely self contained, including its own power supply. Power consumption is estimated at 83 watts total, with 47 watts being required for the logic.

The Input/Output configuration of the Demonstration Unit is shown in Figure 4.3-2. Two analog to digital converters (ADC's) are used on the receiver output because of the small time differential between the two required samples. Timing for the ADC's is derived from clocks provided by the receiver. A one-bit latch is used for the modem input. A separate antenna element weight bus (AE BUS) is driven from the processor I/O Bus. This bus provides data and address lines to the antenna weights with extended driving and timing capability (~1 µsec per transfer) needed to accommodate the remote antenna combining electronics.

The front panel controls of the Demonstration Unit are shown in Figure 4.3-8. Two modes of operation are provided: an operate mode wherein the adaptive steering algorithm is executed; and a setup mode wherein key parameters of the control algorithm are entered into the processor. A calculator type keypad is used to both query the status of the parameters and also to reset the value of the parameter.

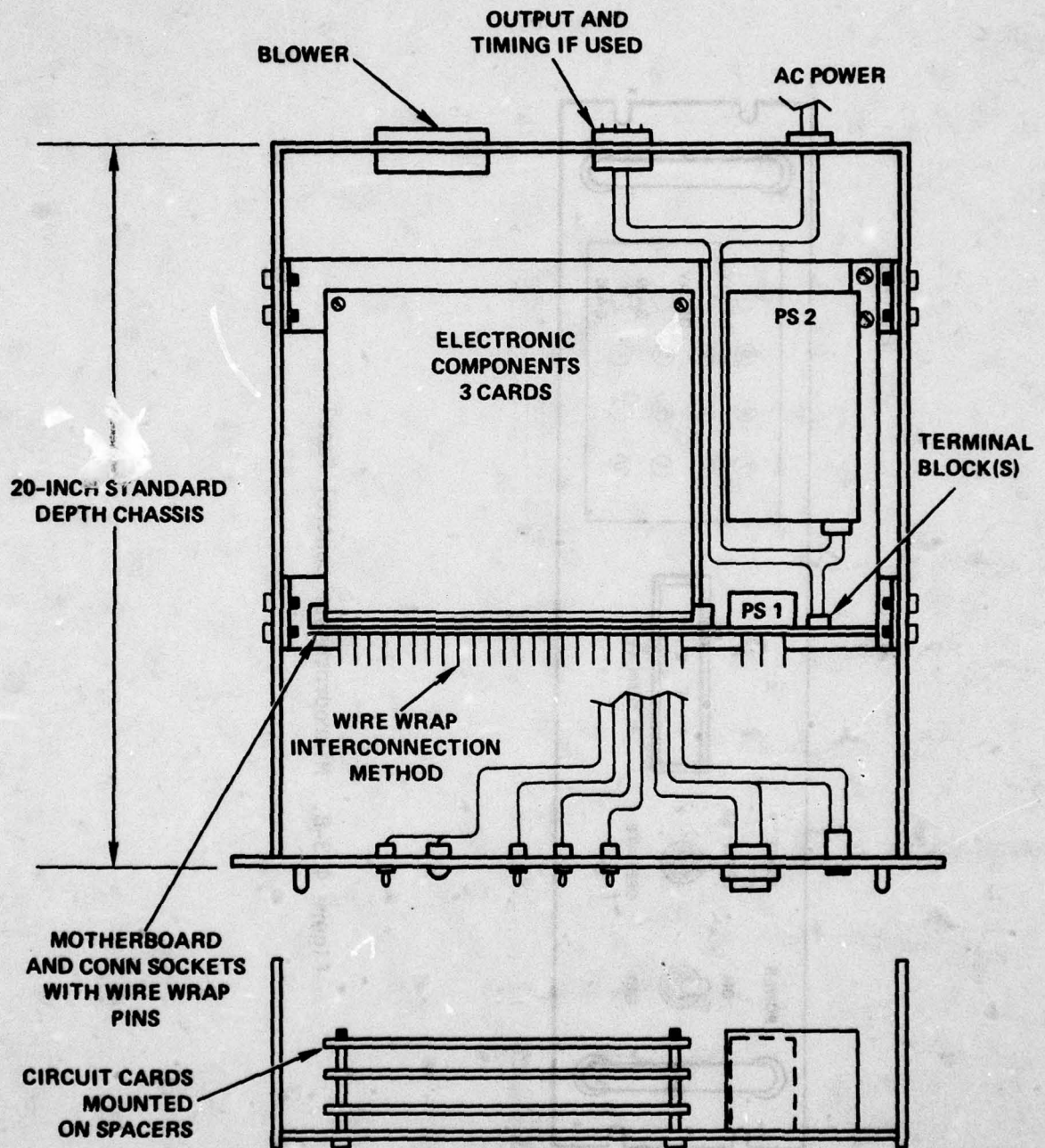


Figure 4.3-7. Physical Diagram of Microprocessor and Related Circuits

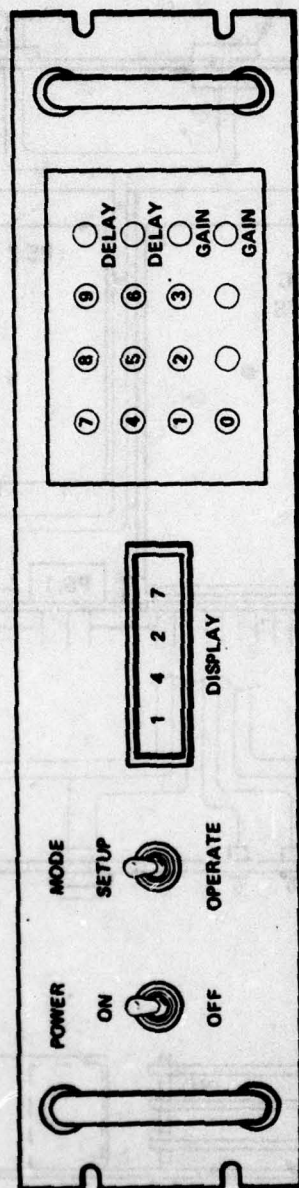


Figure 4.3-8. Microprocessor Control Panel

4.3.3.6.2.1 Firmware Description

The firmware needed to implement the adaptive steering algorithm on the micro signal processor is described in this section. Included in this discussion are descriptions of the firmware as implemented for the simulation study and a recommended implementation for a demonstration unit.

A. Scaling Considerations

A flow diagram of the algorithm as implemented in the simulation study is shown in Figure 4.3-9. For purposes of the simulation the inputs $A(n)$, $B(n)$, and $P(n)$ are obtained from a PDP-11/45 Fortran simulation of the receiver and modem. Similarly, the antenna element weights, $W_j(n)$, are returned to the PDP-11/45 Fortran Simulation. The communications between the PDP-11 and the microprocessor are handled by DEC DR11C Interface. All transfers are in a single precision (16 bit) fixed point format. An implementation utilizing double precision formats in selected places of the algorithm was chosen for the simulation to preserve both the accuracy of the computations and the dynamic range available. In general, the signal flows through the algorithm are maintained in double precision format but the coefficients such as the lowpass filter parameter are maintained in single precision. The two integrators in the algorithm are maintained with double precision accumulation of inputs with the output truncated to 16 bits. With the algorithm scaled in this fashion, no first order problems have become apparent due to quantization.

B. Execution Time

Execution time estimates for the adaptive algorithm are given in Table I. The first entry is the actual execution time encountered in the simulation study without the delaying effect of the Fortran antenna simulation. One iteration of the algorithm is observed to require 52.8 μsec . (four antenna elements, one fixed weight).

A projection of execution time of 20.0 μsec is made if the algorithm could be implemented entirely in single precision arithmetic.

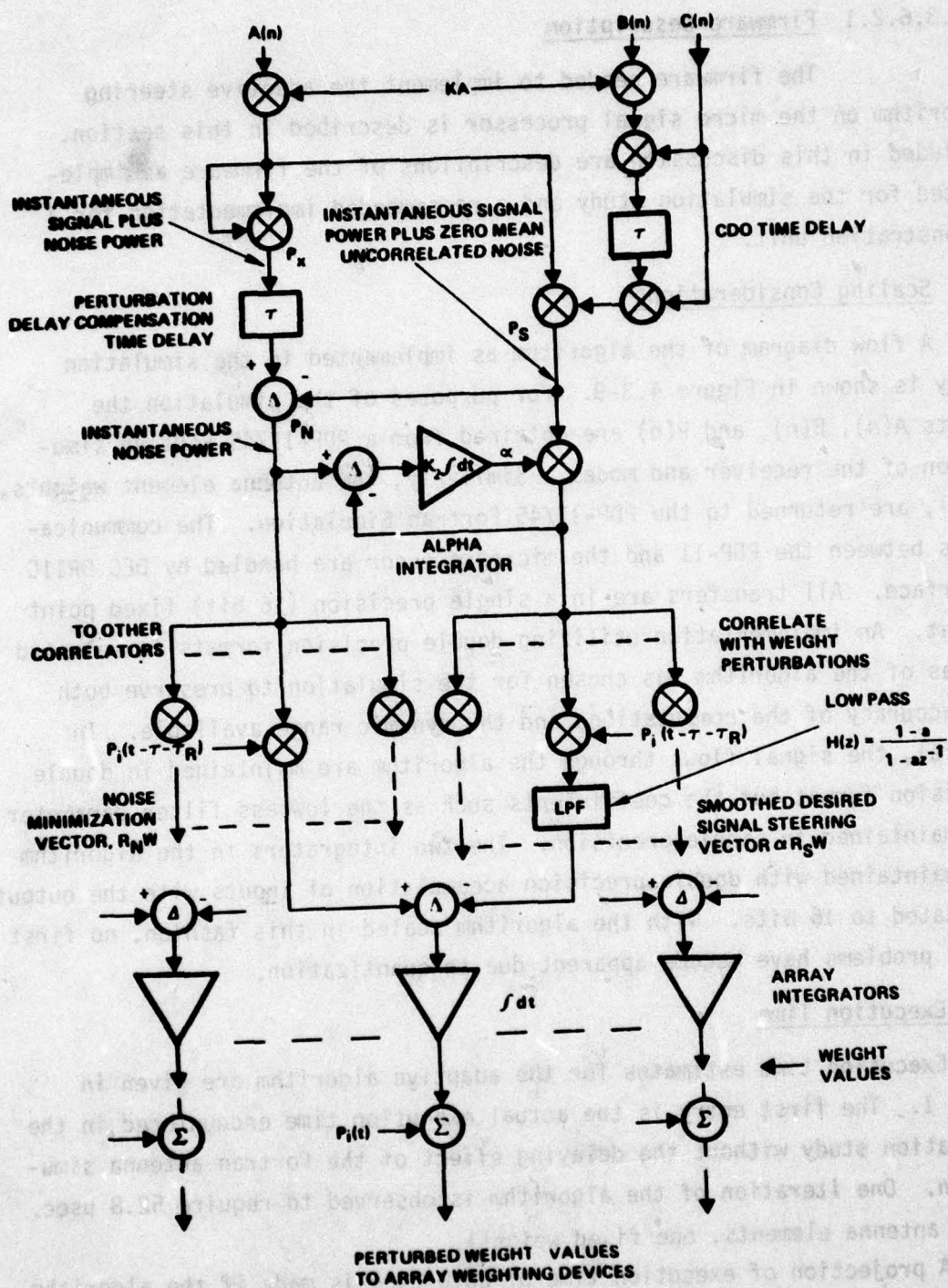


Figure 4.3-9. Microprocessor Flow Diagram

TABLE I. EXECUTION TIMES PER ITERATION

Implementation	Basic Time	Time Per Element	Total (4 Antennas 1 Fixed Weight)
Double Precision	14.4	6.4	52.8 μ sec (actual)
Single Precision	8.0	2.0	20.0 μ sec (estimated)

TABLE II. MEMORY REQUIREMENTS

Data Memory	1.2K x 16	20.4K Bits (RAM)
Instr. Memory	256 x 56	14.3K Bits (ROM)

C. Memory Requirements

Memory requirements can be divided into the two memory areas used in the processor, data memory and instruction memory. Data memory (16 bits wide) requirements are basically driven by length of the two delay elements contained in the algorithm. With a maximum delay length of 500 samples, 1000 locations of writable memory are needed, plus approximately 100 locations for miscellaneous use.

Instruction memory requirements are also modest. A total of 133 lines of microcode are used in the simulation study. For a demonstration unit another 100 lines of code would be required for front panel control functions and initialization.

Total memory requirements for a demonstration unit implementing the adaptive steering algorithm are given in Table II. The requirements are very modest and represent only a small number of integrated circuit packages.

4.3.3.6.2.2 Software Description

A complete listing of the working microprocessor program for the Modified Perturbational PSF Algorithm, including signal recognizer realization is given in Appendix C.

4.3.4 Modem

The intent of this section is not to provide a modem design, but instead to provide a specification of signals required from the modem and to show how a necessary fast sequence synchronization error voltage can be generated.

It is assumed that the desired signal is phase modulated by an a priori discriminant $C(t)$. The modem is responsible for generating this pseudo random sequence as well as acquiring synchronization with it. During the adaptation transient of the array and modem, the modem supplies the microprocessor with its search values of the pseudo random sequence thus enabling the array to adapt when the correct search values are tried even though the modem itself is not locked.

It is noted that only a modestly complicated modem is required in the application addressed by this study. Consequently, the modem may actually be realized in the microprocessor adaptive array controller circuitry. Regardless, the modem will likely require an accurate clock. It is convenient to distribute this clock to the microprocessor in order that the SR time delay be accurately maintained.

The modem also supplies the microprocessor SR function with an estimate of the desired signal's IF output carrier frequency in order to enable computation of the SR phase correction term. As noted previously, this term has only a second-order effect on array/modem performance in that simple searching and acquisition of the desired signal requires trial values of frequency (not instantaneous IF phase), for the SR phase correction term.

If only a slow sequence is transmitted, conventional synchronization circuits all operating upon the receiver's IF output waveform may be applied. However, special treatment is required in the case of the fast sequence which is to be removed at the receiver's RF input.

In order that the modem located as shown in the configurational diagram, Figure 4.3-2, be able to maintain synchronization with the fast spreading sequence $C_f(t)$, it is necessary to provide circuitry which enables measurement of the timing error using signals only available at the output of the receiver. This can be accomplished using the circuits shown in Figure 4.3-10.

In addition to the despreading multiplier already discussed (the multiplier with input $C_f(t)$), an alternative path with its own multiplier is also included. Signals to this path are reduced in amplitude substantially from the main path through use of an RF coupler. If a -10 dB coupler is used, then the main signal is attenuated by approximately 0.1 dB only. The signal supplied to this auxiliary multiplier is the difference between a one chip advanced and a one chip retarded fast sequence multiplied by a perturbational sequence $p(t)$ (the perturbational sequence can be one of the Walsh functions).

The receiver output contains three despread signal terms, the amplitudes of each being respectively proportional to the received

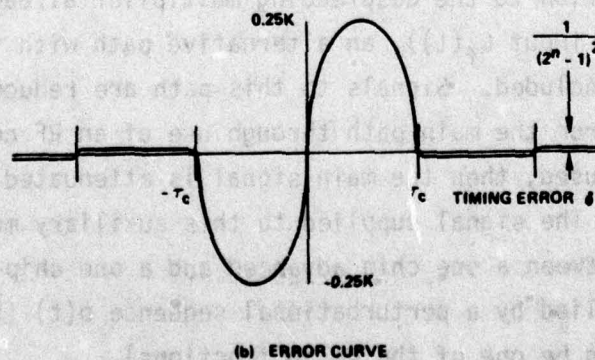
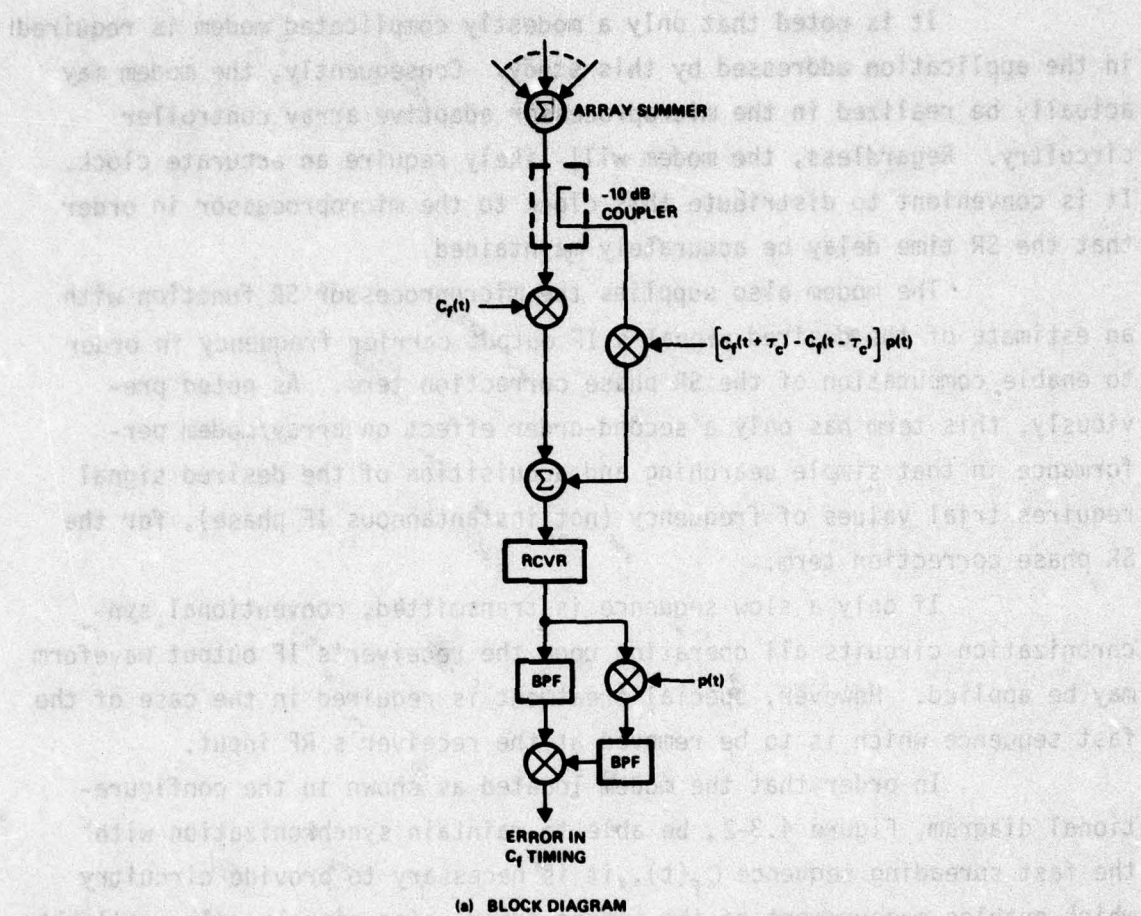


Figure 4.3-10. Method of Enabling Modem Sync to the Fast Spreading Sequence $C_f(t)$

signal's correlation with the advanced, retarded, and principal sequences. The bandpass filters shown are not required if the receiver bandwidth is approximately equal to the bandwidth of the despread signal. Multiplication by $p(t)$ in the right hand path removes this perturbation function from the advanced/retarded correlation term; thus, the result of the multiply following the bandpass filters in the receiver output will be a correlation of the main path correlation with respect to the auxiliary path correlation.

Let us examine this circuit mathematically with the desired signal being represented as follows:

$$S(t) = C_f(t)q(t) . \quad 4-10$$

Assuming a timing error of δ , which for convenience we add into the desired signal terms, the input to the receiver can be expressed as follows:

$$S_{RI}(t) = C_f(t)S(t+\delta) + k\{C_f(t+\tau_c) - C_f(t-\tau_c)\}p(t)S(t+\delta) . \quad 4-11$$

Assuming that the receiver filters to the bandwidth of $q(t)$, it is evident that the receiver output will be proportional to the correlation between the fast sequence contained in the desired signal and the fast sequence generated locally. Designating this correlation as R_{cf} and disregarding the relatively unimportant effect of time delays in $q(t)$, the receiver output, $S_{RO}(t)$, is

$$S_{RO}(t) = Gq(t)\{R_{cf}(\delta) - kp(t)[R_{cf}(\delta-\tau_c) - R_{cf}(\delta+\tau_c)]\} \quad 4-12$$

where G is the receiver's input to output voltage gain. Multiplying $S_{RO}(t)$ by the perturbation $p(t)$, then multiplying this term times S_{RO} and taking expected values yields the timing error $v_e(\delta)$ we have

$$v_e(\delta) = kG^2 E \{q^2(t)\} R_{cf}(\delta) \{R_{cf}(\delta - \tau_c) - R_{cf}(\delta + \tau_c)\} \quad 4-13$$

The pseudo random sequences have triangular autocorrelation functions; therefore, $v_e(\delta)$ will be parabolic. The maximum value can be calculated and is one-fourth of the peak value of the correlation function squared times the RF coupler gain. If we define

$$K = kG^2 E \{q^2(t)\} R_{cf}^2(0) \quad 4-14$$

then the maximum value of the error function is $K/4$. This error curve is illustrated in part (b) of Figure 4.3-10.

It is significant that when the modem is in synchronization with $C_f(t)$ and the error term is near zero, the receiver output is not degraded by the presence of the advanced and retarded sequence terms. This is due to the fact that their autocorrelation functions are zero at this point, and therefore, their contributions are eliminated by the bandpass filtering of the receiver.

4.3.5 Discriminant Function Considerations

In this section, we review those factors considered in the determination of a most suitable discriminant function; specifically we will cover use of the carrier alone, use of uncoded pilot tones, use of multiplicative pseudo random sequences and use of additive pseudo random sequences.

4.3.5.1 Multiplicative Sequence

Since we have chosen a multiplicative sequence for the system discriminant function and for potential spread spectrum AJ processing gain, this topic was discussed in 4.2, Selected System Approach. Here we will briefly review salient advantages of the multiplicative approach then concentrate on the relative merits of fast and slow multiplicative sequences.

The principal reason for choice of a multiplicative discriminant function is that it is easily applied or removed in the process of wideband spectrum spreading. The other approaches either do not allow spectrum spreading or do so only with difficulty.

Secondly, a multiplicative application of the discriminant function enables the entire desired signal to be utilized by the signal recognizer and the S/N maximizing adaptive algorithm.

Third, a multiplicative discriminant is easily applied in upgrading an existing communications network either by a phase shifter in the communications set antenna input/output line or through simple interval hardware changes.

Fourth, a multiplicative sequence is applicable to signals having any arbitrary modulation (AM, FM, FSK, PSK, SSB, etc.).

Fifth, we have proposed the use of a discriminant function which is composed of a slow and fast sequence. Such a dual discriminant is obviously most simply applied and removed if it is multiplicative.

In the following sections, we will review the merits of using either a slow sequence or the product of a slow and fast sequence. Keep in mind that full null steering AJ is obtained in either case since either sequence is postulated to be an unique a priori discriminant.

4.3.5.1.1 Slow Sequence

The principal merit of a slow pseudo random sequence is that synchronization is much easier. Conferencing is facilitated, particularly if the slow code can be received in its entirety by the narrowband radio. In this case, no front-end despread operation is required and thus no noise figure penalty is paid for conferencing. Furthermore, conferencing users can be more widely spaced geographically before the need for individual code tracking loops is required as the chip rate of the pseudo random sequence is increased.

While no spread spectrum AJ processing gain is obtained with only a slow sequence, the 30 to 40 dB null steering AJ is probably adequate for most purposes in the HF/VHF/UHF bands.

Additionally, a slow multiplicative pseudo random sequence is compatible with most unmodified AM radio sets since the sets employ envelope detection. Therefore, the communications system can compatibly accommodate both null steering AJ protected users and unprotected users. This advantage is lost in a frequency modulation communications network since despreading of the AJ waveform is required prior to detection for all users. Another potentially useful feature of the slow pseudo random sequence is that the original frequency division multiplex of the communications system is preserved. A multiplicative sequence having about the same bandwidth as a typical modulation will only slightly increase the bandwidth of the composite signal in an AM communications system since the modulations are orthogonal; thus, the near/far problem often encountered in code division multiplex systems would be avoided here.

The principal disadvantage of a slow pseudo random sequence is that it is more susceptible to sophisticated jamming, in particular, replica jamming. It should be kept in mind, however, that in order to utilize radiated power more effectively than simple CW or noise transmissions, a jammer must reproduce the phase modulations of the pseudo random sequence. Consider a jammer which has the capability of scanning very wide frequency bands and frequency locking upon any transmissions therein. Except for the speed with which this jammer can begin to operate, it is otherwise no more effective than an ordinary jammer since it does not replicate desired signal phase variations.

4.3.5.1.2 Fast Sequence Slow Sequence Product

A fast pseudo random sequence provides both spread bandwidth AJ and additional security against a replica-type jammer, but in an AM communications network, this additional spreading does not significantly contribute security against message intercept.

As noted earlier, the fast sequence is removed at the communications set front-end thus permitting the slow sequence to be received and operated upon by the signal recognizer in the adaptive processor.

Another very important advantage, explained in 4.2, is that jammer eigenvalues are reduced by the spreading ratio when a narrowband receiver is used with a perturbational algorithm as it is in this application. The adaptation dynamics are greatly improved over the full bandwidth correlation approach.

Several penalties must be paid for the use of this fast sequence waveform. First, the wide bandwidth spreading means that an amplitude modulated signal is much more difficult to receive by unmodified radios (although reception is possible). Additionally, the original FDMA properties of the communications network are somewhat compromised in that much more bandwidth is required for a protected user's transmission, thus increasing the probability of a near/far problem. In the case of conferencing, simultaneous despreading at the communications set front end is required for all users. As discussed in Section 4.2.3 this results in a noise figure penalty dependent upon the number of users to be conferenced. Lastly, the synchronization problem is proportionately more difficult, thus a more sophisticated and expensive modem will be required.

4.3.5.2 Additive Code

It is possible to add a discriminant function directly to the information to be transmitted in the communications system. For example, in the case of an AM voice system, a low level pseudo random sequence having about the same bandwidth as the voice spectrum could be added to the microphone signal. In the case of an amplitude modulated waveform, the transmitted signal would have the following representation.

$$S(t) = A[1+m(t)+aC(t)]\cos(\omega_c t). \quad 4-15$$

Expansion of Equation (4-15) reveals that the system waveform can be regarded as the sum of the original amplitude modulated transmission and a biphasic modulated pseudo random sequence having the same carrier

frequency. A signal recognizer in a user's AJ receiver would use this second term as its a priori known discriminant function.

Obviously, such an a priori discriminant function is easily applied, but there are some important drawbacks. Primarily, only a portion of the transmitted signal power can be used at the AJ receiver to generate adaptive array steering commands. In order that the additive function not detract significantly from the amount of power available to transmit information, the discriminant function term must be relatively weak. To an adaptive processor receiving this signal, it is as though a co-located jammer is present with the desired signal. This causes two important performance degradation terms in the adaptive array processor. First, the rate of adaptation to the desired signal is slower in proportion to the amount of power allocated to the discriminant function. Secondly, the presence of the co-aligned jammer (the information content of the desired signal) increases adaptive array weight jitter and requires higher precision in estimation of gradients and adaptive weights. Depending upon adaptive processor offsets (either due to numerical precision problems or to actual device offsets), the array may get a non-optimum solution in its attempt to null the information content portion of the desired signal.

Additionally, the additive code reduces the quality of transmission of the desired signal. To a certain extent, this effect can be alleviated by adding out the code at the receiver. Regardless, amplitude fluctuations, dispersion and timing errors will prevent absolute cancellation of this additive term.

In the case of a very fast pseudo random sequence, the additive technique is probably not practical in that the communications set's transmitter and modulator are probably not designed to accommodate wideband information inputs. In fact, it is likely that a premodulation bandlimiting filter is utilized.

In the case of a frequency modulated communications system waveform, the modulation technique transforms the additive input so that it does not appear as a factorable product in the transmitted waveform.

As a consequence, design of the signal recognizer is difficult at best. As a special case, if the amplitude of the additive pseudo random sequence is properly chosen, a frequency modulator can be adjusted so that the additive modulation input provided a multiplicative term in the transmitted waveform; that is, if the input sequence amplitude and the frequency deviation coefficients are appropriately matched, a transmitted waveform phase shift of either $+\pi$ or $-\pi$ radians could be produced. Then, it is as though the pseudo random sequence were applied with an external mixer or phase shifter. Consider the following frequency modulated waveform.

$$S(t) = A \cos \left[\omega_c t + K_d \left| m(t) + aC(t) \right| \right] . \quad 4-16$$

If $aK_d = \pi$ then we get

$$S(t) = AC(t) \cos \left[\omega_c t + K_d m(t) \right] . \quad 4-17$$

This signal is identical to one using a multiplicative code. Note, however, that phase shifts other than π produce other than biphase modulation and are thus much more difficult to use as discriminant functions.

4.3.5.3 Carrier Alone

As is shown in Chapter 2.0, a modulated waveform's carrier is a sufficient discriminant function if no other signals in the environment simultaneously have that same type of modulation and the same carrier frequency.

Salient advantages of this approach are that all of the waveforms of interest automatically have a discriminant function, and secondly, according to correlation discriminant operator theory, the entire signal power is available for S/N ratio maximization in the adaptive processor.

The obvious disadvantage is that undesirable waveforms also have carriers and could be mistaken for the desired signal. This approach may be workable if the carrier frequency, receiver LO's, etc. are very stable and precisely known. Otherwise, a search procedure is necessary to identify which of the carriers is actually the desired one. Since the presence of a carrier frequency is treated as an a priori discriminant by the CDO, the adaptive processor will maximize any waveform having the right modulation characteristics and the correct carrier frequency. If carrier frequency is searched, the first emitter having the right modulation characteristics and carrier frequency might be an undesired one; therefore, an independent judgment needs to be made either by a human operator or by the introduction of a secondary discriminant function to enable classification of the signal being received optimally. If it is determined that the wrong signal has been selected, then the carrier frequency search is resumed until a desired waveform is found.

If one precisely controls transmitter frequency as well as receiver LO stability, etc., then such a search is unnecessary and only those jammers having the appropriate modulation as well as the appropriate frequency will be treated as desired. As is explained in Chapter 2.0, in order that the adaptive processor treat an undesired signal as desired, it is necessary that the difference in their carrier frequency be in the order of the adaptive processor adaptation bandwidth. In a typical processor, this might be 100 Hz, thus in an HF communications system, the jammer would need to be within 100 Hz of the carrier frequency of the desired signal; perhaps 30 MHz. This in turn, means that the jammer frequency would need to be set to within roughly 30 ppm in order that it be treated as a desired signal by the adaptive processor.

If the waveform carrier is being used as a discriminant function, then it is necessary that the signal recognizer be provided with this information. This, in turn, implies the need for either a tracking loop or a reference frequency. Since similar circuits are required in the case of a pseudo random discriminant function, use of

the carrier alone merely saves circuitry which generates and synchronizes the pseudo random function. At least in the case of a slow pseudo random sequence this additional circuitry is not highly complex. This means that the carrier tracking approach is almost as difficult to implement as the much more secure narrowband pseudo random discriminant approach.

4.3.5.4 Uncoded Pilot

This technique requires precisely the same circuitry for acquisition and tracking as does the carrier discriminant approach. Furthermore, it can be seen that the uncoded pilot is a special case of the additive code just discussed in Section 4.3.5.2, the difference being that a CW is transmitted in place of a pseudo random sequence. Therefore, all of the problems of that approach are incurred without compensating benefits. Consequently, use of an uncoded pilot is regarded as a very poor alternative.

5.0 SIMULATION/EXPERIMENT

Concepts presented in the preceding three chapters which displayed promise were evaluated both independently and in conjunction with other concepts by means of computer simulation. Part of this simulation, referred to as the "experiment", involved the realization of a complete adaptive processor with signal recognizer in microprocessor circuitry. The microprocessor was, in turn, interfaced with a general purpose digital computer simulating an electromagnetic environment, antenna array, receiver and analog-to-digital interface. In effect, the microprocessor was controlling an adaptive array connected to a generic receiver.

For reasons that are made clear later, the general purpose digital computer also simulated the microprocessor in a different set of experiments.

Basically, this chapter is organized into three main sections. The first, 5.1, gives a description of the methods incorporated into the electromagnetic environment/antenna array/receiver simulation. At times, it is convenient to use frequency domain analysis and at other times, a time domain approach. This section explains where and how domain transformations can be accomplished and significant limitations of the approximate methods actually used.

The second major portion of this chapter, Section 5.2, is a detailed review of the microprocessor portion of the experiment. Hardware details, algorithm description, advantages and disadvantages of the approach used, and problems encountered are discussed. Organization of this section is according to the microprocessor device being utilized, the 8080A, the AMD 2900 or the Slash 4 microprocessor simulator.

Section 5.3 is the last major portion of this chapter. Here we give experimental simulation results. In addition to providing an illustration of the concepts discussed in Chapters 2.0, 3.0 and 4.0, this section also serves to provide engineering design data.

The following topics are among those investigated and presented: The affect of varying the perturbational sequence amplitude (for achieving optimal tradeoffs between accurate gradients and weight jitter degradation

of output signal-to-noise ratio), continual versus periodic weight update, the affect of algorithm gain parameter adjustment, the affect of the number of bits used in both the array weight and in the receiver output A/D converter, the affect of varying the chip rate of the a priori discriminant function, ability of the adaptive processor to maintain performance as the adaptive array is subjected to rotational motion and demonstration of instabilities which result when certain algorithm parameters are incorrectly chosen.

5.1 Simulation Methods

Figure 5.1-1 illustrates the conceptual organization of the simulation. Observe on a large scale basis that the algorithm, realized in the microprocessor, is interfaced to a simulation computer. This computer generates waveforms attributed to the desired signal and jammers, the antenna array, the receiver and the analog-to-digital interface between the receiver and the microprocessor. As indicated, it is at times convenient to use different domains to represent waveforms in this system. For example, signals, jammers, adaptive weights and the receiver output are most conveniently represented in the time domain. Alternatively, mutual coupling among the antenna elements and bandpass characteristics of the assumed receiver are conveniently treated in the frequency domain.

This simulation could be treated exactly according to the flow diagram in Figure 5.1-2. The desired signals, coherent jammers and thermal noise at the antenna input are conveniently generated as a function of time. Specifically, we use a pseudo random generator to generate an envelope function for a desired signal, and a random number generator to provide envelopes for jammers and thermal noise. Carrier frequency terms are assumed through the use of complex arithmetic and applied later as a real modulated function at the receiver IF output. All emitters are assumed to have the same carrier frequency. A fast Fourier Transform (FFT) converts these inputs to the frequency domain, facilitating application of the antenna array mutual coupling matrix and/or dispersion effects in the RF circuitry. It is difficult to treat mutual coupling

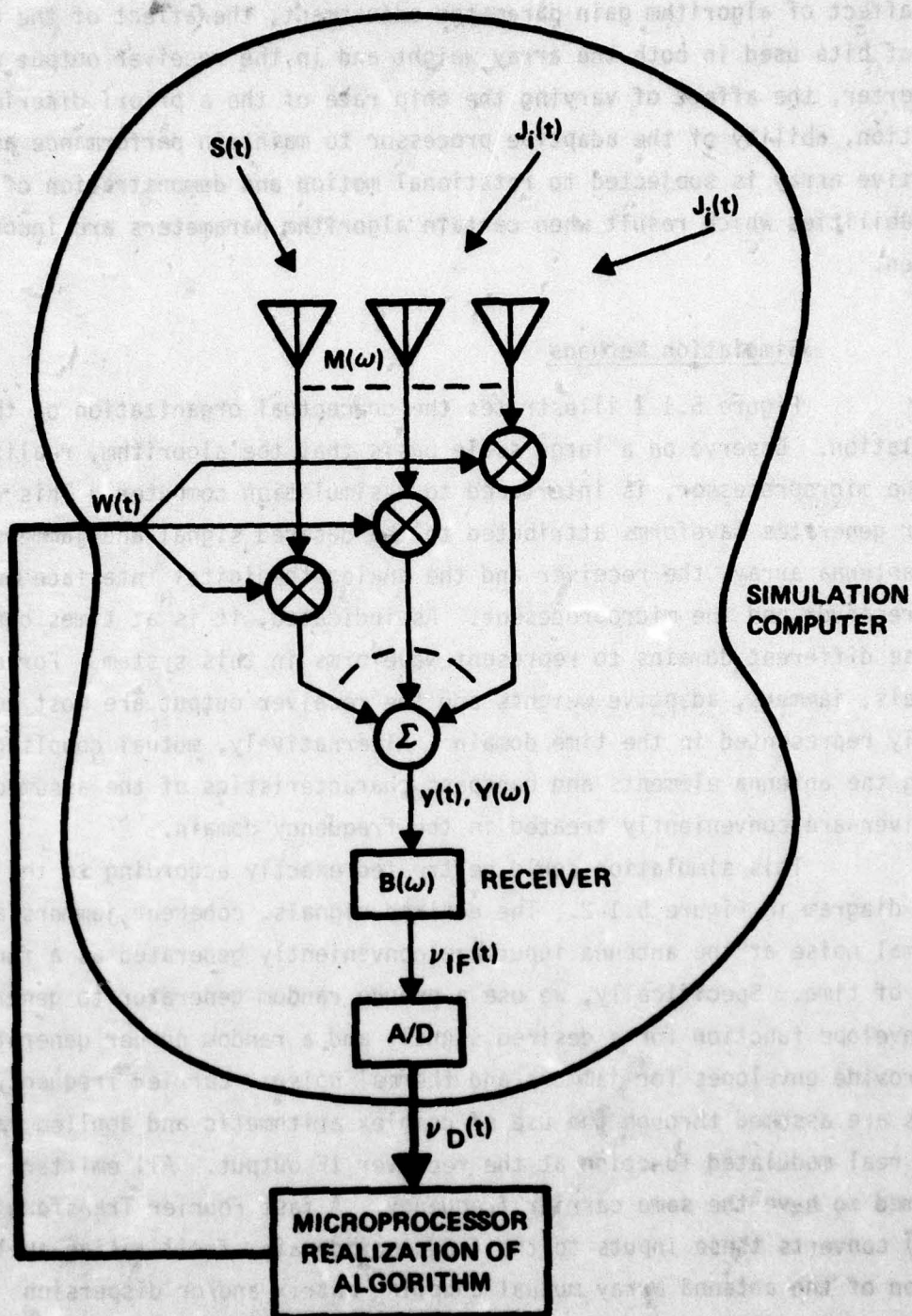


Figure 5.1-1. Block Diagram Showing Conceptual Interface of the Simulation Computer with the Microprocessor

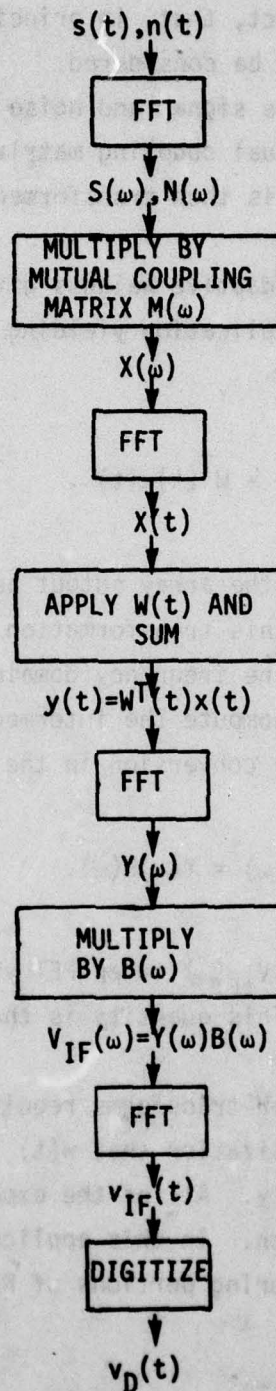


Figure 5.1-2. Idealized Simulation Flow Diagram

in the time domain due to the fact, that, in principle, an infinite number of multipath terms would have to be considered.

Multiplication of the signal and noise inputs in their frequency domain representation by the mutual coupling matrix $M(\omega)$ yields the antenna input voltage vector $X(\omega)$ which is then transformed with the FFT to get the conventional input $x(t)$.

At this point, the adaptive weights given as a function of time are easily applied by multiplication yielding the array output $y(t)$, we have

$$y(t) = W^T(t)x(t) . \quad 5-1$$

Again, applying the FFT, we get the array output as a function of frequency, $Y(\omega)$. It is desirable to make this transformation because the receiver is most conveniently described in the frequency domain. Taking the receiver description to be $B(\omega)$, we can compute the intermediate frequency output (assuming an arbitrary frequency conversion in the receiver) as follows:

$$V_{IF}(\omega) = Y(\omega)B(\omega) . \quad 5-2$$

Finally, subjecting $V_{IF}(\omega)$ to an FFT yields the receiver IF output as a function of time. This quantity is then digitized and output to the microprocessor.

The number of Fourier transforms required can be reduced by a factor of two through the realization that $w(t)$ is almost always representable as quasi-static quantity. All of the expected value equations, for example, make this assumption. In this application, it is not possible for W to change significantly during portions of RF or IF cycles, therefore we may write:

$$Y(\omega) \approx W^T X(\omega) \quad 5-3$$

where we are treating W as a constant.

Additional approximations were made during the course of this study. Since we were primarily interested in application of the technology to narrowband communication signals, dispersion effects due to mutual coupling are entirely negligible. In the future, however, particularly if considerable spectrum spreading is utilized, it will be desirable to allow for mutual coupling effects. Given that mutual coupling is negligible, we may eliminate the first FFT operation and obtain the array output $y(t)$ exactly.

A more important approximation is made regarding the treatment of the receiver. In order to perform a very accurate simulation, it is necessary to know details of the receiver amplitude and phase characteristics. This is especially important when one attempts to adapt the array at the maximum possible rate. Specifically, the perturbational terms varying most rapidly will be differentially delayed with respect to the more slowly varying perturbational terms, thus affecting the accuracy to which the gradient can be computed. Therefore, we initially provided in our simulation for the FFT and its inverse at this point so as to enable an accurate receiver bandpass limitation simulation.

Alternatively, the mid-band time delay of the receiver is probably the single most important aspect of the receiver filter characteristics in that this quantity determines the time delay associated with passing information through the receiver. Consequently this parameter strongly affects algorithm stability as maximum adaptation speeds are approached. Since specific receiver details are unknown and since we were primarily interested in establishing absolute limits of the perturbational procedure, the work reported in 5.3 has treated the receiver bandpass simply as a time delay.

Under this assumption, the FFT and its inverse are easily taken since we have

$$v_{IF}(t) = F\{Y(\omega)B(\omega)\} = F\{Y(\omega)e^{-j\omega\tau}\} = y(t-\tau) . \quad 5-4$$

It is important that future simulation work include a refinement of this assumption, specifically providing calculations for a generic receiver bandpass characteristics. The parameter of greatest interest would be accuracy of gradient computation as the bandwidth of the perturbational sequences approaches the bandwidth of the receiver. Until these simulations are performed, it is recommended that a factor of between 2 and 10 reduction in the perturbational bandwidth with respect to the receiver bandwidth be employed.

Since the $w(t)$ are utilized in a narrowband array, it is convenient to represent these quantities as phasors. Consequently, signal and noise are represented as phasors multiplied by a real time varying envelope. We write the i^{th} waveform as

$$\hat{S}_i(t) = e_i(t)e^{j\omega t} A_i e^{j\phi_i} \quad 5-5$$

where the symbol " \wedge " means complex envelope. Since we presume an arbitrary frequency conversion from the RF input of the receiver to its IF output, we do not need to specify the carrier frequency of these complex quantities until one of the last steps in this simulation. Using the complex notation, the IF output will be computed as a phasor having a real envelope function of time. We have

$$\hat{V}_{IF}(t) = e^{j\omega_{IF}(t)} \left\{ e_{IF}(t) e^{j\phi_{IF}(t)} \right\} = e^{j\omega_{IF}t} \left\{ \sum_i e_i(t) \sum_k W_k e^{j\phi_{ik}} \right\} \quad 5-6$$

where ω_{IF} is the receiver's final IF center frequency and k designates an antenna input. We move the complex envelope description to real time as follows:

$$v_{IF}(t) = e_{IF}(t) \cos(\omega_{IF}t + \phi_{IF}(t)) \quad 5-7$$

In the simulations reported here, the IF frequency carrier was chosen arbitrarily but with the restriction that it not be a harmonic or otherwise correlate with terms present in the simulation.

5.2 Microprocessor Experiments

Three different experiments, two using actual microprocessors and the third using a simulated microprocessor, were conducted. Description of these experiments is divided accordingly into three sections with each section being organized around the microprocessor. In each case, there is a description of the hardware used in the experiment, the algorithm realized by the hardware and a brief summary of results obtained. (Comprehensive results are presented in 5.3.)

In all cases, the electromagnetic environment/antenna array receiver/interface was simulated as described in Section 5.1 by a general purpose digital computer. In each case, a different general purpose computer was used, this factor being controlled mostly by convenience in achieving the actual interface of the digital computer with the microprocessor.

Section 5.2.1 is devoted to a summary of the Intel 8080A-based microprocessor circuitry. Section 5.2.2 is a summary of the AMD 2900-based device, and Section 5.2.3 summarizes what we refer to as the "Fortran microprocessor", a simulation of a generic microprocessor also residing in the general purpose digital computer.

5.2.1 Intel 8080A Experiment

Early in this study while algorithm concepts were still in a state of flux, a simple 8080-based microprocessor was selected. It was discovered, for reasons to be discussed shortly, that this hardware was inadequate. This phase of the study was regarded as being very useful in that we were able to establish roughly the number and types of computations that would be required, the frequency with which the microprocessor program would need to be changed, and how much real time would be needed by the combination microprocessor simulation computer to achieve a

corresponding simulation time. Most of the simulation results obtained in this phase were negative, but the difficulties encountered were in retrospect important factors in subsequent successful experiments.

5.2.1.1 Hardware

The heart of the microprocessor circuitry was an Intel 8080A. This device was selected initially for two reasons. First, at the time it was selected, it was the only MIL spec microprocessor available. Second, developmental facilities were readily available.

The complete "8080 machine" contained the 8080 microprocessor and a 4K programmable read-only memory, holding the standard 1K Intel monitor plus the program which performed the algorithm. The machine also contained 1K of random access memory which the program could use to store tables and intermediate results.

The developmental facility consisted of a Tektronics microprocessor lab with a microprogramming language, two floppy discs for program storage, and a device for programming programmable read-only memory integrated circuits.

The 8080 is an 8-bit microprocessor. In order that 12 bits of accuracy be maintained, the 8080 was operated in a double precision mode. A 12-bit accuracy goal was selected because 12-bit analog-to-digital and digital-to-analog converters are readily available. With 16-bit precision in the microprocessor, it was expected that the A/D and D/A interfaces would represent limiting factors in algorithm numerical capability.

The 12 bits used internally in the microprocessor allow the representation of voltage or weight quantities to one part in 10^3 . This in turn implies a power dynamic range of 60 dB. Power data was carried as a 12-bit number assumed to be positive. Weight data, on the other hand, were assumed to be in a range of -1 to +1 and carried as two's complement, the 12th bit being a sign bit.

Since the 8080 is an 8-bit machine and the 9600 baud line handled 8-bit quantities, two words were required to convey one weight

value. The "left over" four bits conveyed address information or data requests to the receiver simulation. If the four control bits were all one, the receiver was commanded to supply a reading of its analog-to-digital converter outputs. Alternatively, these bits represented the number of the element whose weight was being updated.

The environment/array/receiver simulation program was run on a Datacraft 6024/5 minicomputer. This is a 24-bit machine having 24K words of memory. It was connected to a line printer, card reader and teletype. While there was also a disc drive, this device was used only to hold the operating system. It was necessary to read in user programs through a card reader, and a complete compilation was required for each run. This limitation was ultimately serious in that it contributed substantially to the amount of time required in performing a given set of simulation runs. All commands to the simulation were read in from the card reader, and all results of the simulation were output to the line printer. The teletype was used only for system commands to the Datacraft computer.

A Special Handler had to be written for this computer/microprocessor configuration in order to allow the 8080 to drive the Fortran simulation just as it would drive a real set of array weights.

Interface of the Datacraft computer with the 8080A microprocessor was achieved through a 9600 baud serial line. Considering the amount of data transfers required in both directions, this was an inadequate data link in that it imposed a serious bottleneck resulting ultimately in excessive amounts of laboratory time required to simulate a real-time experiment.

5.2.1.2 Algorithms

Two algorithms were employed in this situation: simple suppression and S/N maximization based on the log power technique described in Section 3.3.4. The suppression algorithm was included as an initial effort due to its simplicity and well-known performance. Thus, it was useful in establishing that the microprocessor hardware was in fact working properly.

At this stage in the study, it was not yet realized that an algorithm utilizing "instantaneous" power was achievable. Consequently, the power detector assumed utilized a lowpass filter for obtaining an average power. Basically, the algorithm is as described in Section 3.1.2.1, except that no desired signal power detector was used.

Later a signal-to-noise ratio maximizing algorithm, the log power technique described in Section 3.3.4 was simulated. A principal reason for selection of this algorithm was that no multiplications were required as they were inherent in the additions and subtractions of the appropriate logarithms. Since the 8080 does not do automatic multiplication or division and since hardware devices are available, such as log power amplifiers, for generating the necessary input functions, this approach promised the realization of a relatively sophisticated algorithm by a relatively simple microprocessor.

This early microprocessor program did not incorporate a signal recognizer. Instead information on signal power was obtained directly from the receiver simulation digital computer.

Perturbational sequences for these algorithms were the three level sequences described in Section 3.2.2.3. Specifically, each weight was successively perturbed and changes in total output power and signal output power were used to compute a partial derivative of power change with respect to a given weight. As discussed in Section 3.3.4, these partial derivatives can be converted to the desired gradient. Another advantage of the three level sequence is that it is very simple to implement in that no correlation technique for weight extraction is required, nor is a ROM table of perturbation functions. Alternatively, this approach is not suitable when modulation jammers will be present. Thus, its use here was appropriate only in the initial development phase.

5.2.1.3 Results

Simulation results of this experiment were disturbing. While the simple suppression algorithm worked adequately, the S/N maximization algorithm was unsatisfactory. Regardless, a great deal was learned about

AD-A068 890

HARRIS CORP MELBOURNE FLA

F/G 17/4

APPLICATION OF A CORRELATION DISCRIMINANT OPERATOR TO PERTURBAT--ETC(U)

APR 79 G P MARTIN

F30602-77-C-0073

UNCLASSIFIED

RADC-TR-79-44

NL

3 OF 4
ADA
068890



what would be required in order to achieve successful algorithms in terms of both microprocessor circuitry and laboratory facilities.

The 8080 was determined to be far too slow for use in a final design. An elapsed microprocessor time of 775 microseconds was required for one weight component update. At this rate, the hardware connected to a real receiver would take about one second to suppress two jammers by 10 dB. More disturbing, the signal-to-noise ratio maximizing algorithm slowly worked to obtain about an 18 dB S/N but failed to hold this value, ultimately stabilizing at about -2 dB S/N. Because it was so hard to modify the microprocessor program, the cause of this behavior was not investigated. Some possibilities are an erroneous program, lack of sufficient precision for this algorithm, differential errors in the log power evaluations or biases due to the $\ln(1+\delta)$ approximation. The adaptation transient is suggestive of a steering error bias.

The most important result of this set of experiments was the practical experience gained and procedural details learned regarding developmental work of this type. First, a faster data link was essential. The 9600 baud data link so delayed the computations that time periods of 15 and 20 minutes were required in order to simulate an algorithm running for only one second. This alone prevented any accurate examination of long-term algorithm behavior.

Secondly, it was essential that an easier means of microprocessor program modification be available. Although it was not hard to change the microcode program through use of the Tektronics lab equipment, getting the change into the hardware and the hardware into the lab was more difficult. Comparatively, a microprocessor program change was about as difficult as an analog circuit hardware change.

Thirdly, microprocessor circuitry capable of achieving rapid multiplies would be required. Such multiplication enables the realization of the much more sophisticated "modified perturbational PSF algorithm", an algorithm too complex for the simple 8080.

Finally, a machine at least an order of magnitude faster on simple operations such as addition and subtraction was needed in order to achieve a reasonably fast algorithm.

Concerning the simulation computer and the interface between it and the microprocessor, a disc storage of the compiled simulation program was desirable as such compilation time could be a substantial portion of the time required for a given set of simulation runs.

5.2.2 AMD 2900 Experiment

Most of the problem areas encountered in the previous simulation experiment were corrected during this experiment. Additionally, a sophisticated algorithm was performed and good results obtained. As in the previous section, we describe the hardware first, then the algorithm, and finally the results obtained.

5.2.2.1 Hardware

Because of algorithm execution speed requirements, a bipolar bit slice microprocessor was selected. This class of microprocessor is capable of instruction execution times on the order of 200 nanoseconds; in contrast, the 8080 requires about 2 microseconds. The device selected is from the family of the AMD 2900 integrated circuits. Additionally, the device is interfaced with a single chip high speed multiplier, TRW-MPY16, which performs a full 16 by 16 multiply obtaining a 32-bit rounded product in 200 nanoseconds. The high speed multiplier greatly enhances the microprocessor's signal processing capability.

This microprocessor is described in detail in Section 4.3.3.6, Selected Microprocessor Design. Essentially, the experimental microprocessor circuitry is the same as that specified for use in the system design.

A different antenna/receiver simulation computer, a PDP-11/45 having a 16-bit word size and a 64K memory, was used during these experiments. The PDP-11/45 minicomputer was connected to a line printer, a card reader, a tape drive, a teletype and importantly, a disc file mass storage. A great advantage of this configuration was that editing could be done utilizing the disc and furthermore compiled programs could be stored on disc.

Another very important factor is that the 2900 microprocessor was hardwired to the PDP-11/45. This facilitated microprocessor program changes as well as data transfer during a simulation run. From experience gained during the first experiment, it was concluded that an easy method of modifying the microprogram was essential. This hardware configuration provided that facility in that a direct memory access link existed between the PDP-11 and the AMD 2900. The DMA link allowed assembly of the program for the 2900 in the PDP-11 using a cross assembler developed at Harris, and subsequent downloading of this program into the control memory of the 2900. Furthermore, data could be loaded directly into the 2900 memory, particularly tables which would be in ROM memory in an actual application.

Importantly, the DMA link between the PDP-11 and the AMD 2900 was orders of magnitude faster for data transfer than the one used in the previous experiment. Thus, the amount of laboratory time required to simulate a given amount of algorithm time was greatly reduced.

5.2.2.2 Algorithm

The modified perturbational PSF algorithm described conceptually in Section 3.3.1 was selected. Since the experimental circuit was selected for use in the null steering AJ system, a comprehensive summary of this algorithm as realized in the AMD 2900 is given in 4.3.3.

5.2.2.3 Results

This algorithm as implemented on the AMD 2900 microprocessor achieved a complete iteration cycle in 52.8 microseconds. This speed obtains a 20 dB signal-to-noise ratio improvement using a 4-element antenna array in less than 15 milliseconds. Although adaptation at this rate is roughly a factor of three slower than the maximum rate based on restrictions imposed by receiver bandpass filters, it is nevertheless a very practical processor. It is obviously a significant improvement over the earlier results in speed as well as sophistication.

We note that the actual time required in the laboratory to produce these results was about two orders of magnitude longer than the microprocessor computation time due to overhead imposed by the Fortran environment/receiver simulation program. Regardless, this simulation was sufficiently fast that adaptation to the desired signal could be seen in a reasonable amount of time.

In order to ensure that an accurate simulation had been realized, the results of the adaptation were carefully matched against Fortran simulations run on the Slash 4 computer. Step-by-step results matched very closely with only slight deviations along the path and with convergence to the same final value. An example comparison is given in Appendix B. Deviations are due to differences in precision of the two machines.

Even though the AMD 2900 and the PDP-11 represented a great improvement over the first experimental configuration, it nevertheless remained difficult to achieve a large amount of simulation data in a convenient form. Independently, a computer program was developed for the Slash 4 which simulated the entire environment array receiver microprocessor configuration. The bulk of our AMD 2900 experimental investigation, therefore, went to establish complete agreement between experiment and the Slash 4 simulation. Thus, the real value of the AMD 2900 simulation was in proving that a microprocessor and its related circuitry could be realized in hardware and could run the algorithm at an acceptable rate.

5.2.3 Fortran Microprocessor Experiment

Probably our most important microprocessor development tool was a microprocessor program simulator written for the Datacraft Slash 4 computer. This "Fortran microprocessor" was the basis against which all the other experiments were compared. Furthermore, it was the most sophisticated, the easiest to develop, the most flexible and the most accurate of the simulations. This simulation was used in three different modes. First, it was used to develop an algorithm for implementation in

a real microprocessor, then it was used to check the real microprocessor's accuracy, particularly to ensure that the real microprocessor code was accurately written, and last it was used to obtain bulk data predicting the actual microprocessor's performance. In this last mode, it was possible to obtain orders of magnitude more performance data in a better form for analysis and review than from the actual microprocessor.

There are several advantages of running such a microprocessor simulation on a time sharing computer. First, the simulation can be written in Fortran, by far the easiest language to use. Secondly, the floating point implementation of the algorithm can isolate precision problems from algorithm conceptual problems. Effects of bit precision can be introduced at a later stage. Additionally, any part of the operation of the algorithm can be examined by inserting output statements and displaying intermediate results. When problems are discovered, it is very easy to modify the algorithm with a simple Fortran statement change.

The fact that the Slash 4 computer supported Tektronics graphics terminals was also of great use in that plots of results conveyed a great deal of information very quickly, much as would a laboratory oscilloscope.

Naturally, the environment/antenna/receiver simulation was run on the Slash 4 computer simultaneously with the "Fortran microprocessor". In this way, very rapid development of a microprocessor program was obtained. Once a promising design was established and refined, it could then be programmed in the lower level language used by the actual microprocessor.

5.2.3.1 Hardware

The Fortran microprocessor simulation was developed on a local timesharing system using a Datacraft Slash 4 computer running under the Vulcan operating system. This is a time sharing operating system which allows great flexibility in programming and substantially improved access to the computer. The Slash 4 computer has 64K 24-bit words partitioned into two sections of memory, but being a virtual memory

operating system, the effective program size is unlimited. It supports line printers, card reader, scope terminals, graphic terminals, discs for the storage of the user programs and magnetic tape. High speed input/output terminals such as the scope terminal and graphics terminal allow for very rapid output of simulation results, particularly useful for troubleshooting and debugging. Furthermore, access could be gained to the computer almost any time of the night or day.

5.2.3.2 Algorithms

All of the algorithms used in any of the experiments were implemented on the Slash 4. Although these simulations were realized with floating point arithmetic and had either equal or greater accuracy than the experimental simulations, through the use of a special precision truncation program it was possible to simulate an arbitrary number of bits precision in both input and output.

The most important algorithm simulated and the one for which extensive data has been generated is the perturbational PSF algorithm. This algorithm, of course, is the one implemented on the AMD 2900 microprocessor and which is recommended in Chapter 4.0, AJ System Design.

Algorithms implemented on the "Fortran microprocessor" were verified analytically in a number of ways. First, gradients for a particular weight value were calculated "by hand" and compared with the "microprocessor" obtained results. Long term trends were calculated by a very general, previously existing expected value closed form program based on an eigenvalue analysis (EVA).

5.2.3.3 Results

The verified Slash 4 microprocessor algorithm was used to verify the results of the microprocessor hardware implementations. Then operating in the reverse mode, it was possible to effectively "simulate" the simulations. Consequently, the "Fortran microprocessor" was a very powerful tool, first providing a rapid and accurate means for developing actual microprocessor programs, and then, providing a very

convenient and rapid means for simulating the developed microprocessor. A summary of simulation results obtained follows.

5.3 Simulation Results

All of these simulation results were obtained from the Fortran microprocessor simulation experiment. As explained in Section 5.2, results obtained here are representative of the actual AMD 2900 microprocessor. Consequently, these results are also considered to be representative of the proposed system design given in Section 4.3.3.

The following is a list of topics covered in this section:

5.3.1 Standard Case - Here we identify the standard case from which all parameter variations are made. Additionally, a theoretically calculated adaptation rate is compared with the actually obtained adaptation transient. Comparisons are also made for the case of different perturbational sequence sample periods, SP.

5.3.2 Array Gain - The principal array gain parameter, KA, is varied.

5.3.3 Perturbation Sequence Effects - In this section, perturbational sequence amplitude is varied showing the effect of gradient precision as well as signal-to-noise ratio degradation due to weight jitter, continuous versus periodic updating of the weights, and the effect of perturbation sequence length.

5.3.4 Alpha Loop Gain - Gain of the signal-to-noise ratio estimator in the PSF algorithm is varied in conjunction with bandwidth of the desired signal steering vector filter.

5.3.5 Signal Recognizer Parameters - Here we examine effects of varying the rates of the desired

signal a priori discriminant and delay in the signal recognizer.

5.3.6 AGC Effects - The effect of including receiver AGC on the adaptation transient is shown.

5.3.7 Digital Word Size - The effect of reducing the number of bits both in the weight description and in the receiver output A/D converter is demonstrated.

5.3.8 Rotation Effects - The adaptive array is rotated in the plane of the desired signal and jammer at varying rates. The ability to maintain a good S/N is demonstrated.

5.3.9 Two Jammers - Cases for two jammers and a single desired signal are investigated.

5.3.10 Instabilities - The effect of improper choice of algorithm parameters is shown.

5.3.11 Signal Power - The effect on adaptation transients due to desired signal power variation.

5.3.1 Standard Case

An adaptation transient for our standard case parameters is given in Figure 5.3-1. Practically all of the results to be presented in this section will have the format illustrated in this figure. The ordinate is identified as signal-to-noise ratio or relative power in dB. Since signal-to-noise ratio is a redundant parameter and one which tended to clutter the graphs, this parameter is generally omitted unless otherwise stated. In the usual case, there are two curves appearing on every plot. One is total noise power with respect to thermal noise in the receiver output and the other is desired signal power with respect to thermal noise in the receiver output. In all of the simulation runs presented here, no averaging is done and only instantaneous quantities are plotted. Comparison of these results with expected value results will require an averaging of the curve shown.

In all cases, desired signal power is less than jammer power. Furthermore, desired signal power is almost always a slowly varying function near the center of a graph. It is identified by the typed symbol S. Noise power is identified by the symbol N. Observe that input N/S is available from the graph at time zero, since an initial omni pattern is always assumed. In all of the data presented here, the time is given in milliseconds.

A summary of important algorithm parameters is given at the right hand edge of each graph. We now identify these parameters and give their values for the standard case. Please refer to Figure 5.3-1.

The first three parameters listed are KA, KH, KP. Respectively, these values are algorithm gain parameters. KA is the square root of the array integrator gain K_1 as is identified in Section 4.3.3. KA is applied as a square root of array gain because it is convenient to multiply this parameter times the sampled receiver output waveform as illustrated in Figure 4.3-9, and thus avoid the necessity of N multiplies at the array integrators. Here, KA=0.01, therefore $K_1=0.0001$.

The term KH is the gain parameter for the alpha loop (signal-to-noise ratio estimator in the PSF algorithm). The term K_2 in Figure 4.3-9 is equal to the product $KH KA^2$.

Amplitude of the weight perturbation sequences is given by KP. This parameter is expressed as a fraction of the largest weight value allowed. Here, KP=0.001.

The next set of parameters, located between two horizontal rows of asterisks, have the units of time or inverse time. They are BF, SP, CT and PP.

BF represents the width of the lowpass filters in the desired signal steering vector estimation loops. This parameter is 100 hertz in the standard example.

A very basic algorithm parameter is the period of a perturbational chip. This is designated as SP, for "sample period". In an actual design, SP would be determined either by receiver bandwidth or by the maximum rate at which the digital circuitry could perturb the

weights. Regarding absolute maximum rates, the 20 microsecond period given here would require a receiver bandwidth in excess of 25 kilohertz.

The parameter CT is the signal recognizer delay. The acronym CT is derived from correlation time. Note that on the computer-generated plot, CT is given as an integer times SP. An integer relationship simplified program computations without significantly reducing program generality. Note that in this case a signal recognizer delay of 660 microseconds is simulated.

PP is the perturbation sequence period. Since we are using Walsh functions as explained in Section 3.2, sequences having length 2, 4, 8, 16, 32, etc. are used depending upon the number of weighted inputs. Observe that in this case, the period is 33 SP rather than 32 SP. In order to observe array output in the absence of weight perturbations (so as to determine the highest quality of output obtained), the Walsh function sequences were modified by adding a vector of all zeros to the standard 32 term sequence. Consequently, when PP=33 SP, plotted results show an unperturbed array output.

Under the heading "signal" are the parameters IF and CT. Respectively, these are the receiver's intermediate frequency output and the chip period of the desired signal a priori discriminant.

As explained in Chapter 4.0, the sampling procedure in effect performs a down conversion of the receiver's intermediate frequency to near baseband. In order to work with numbers having a convenient size, we arbitrarily select a frequency between zero and the width of the receiver's bandpass filter; in this case, a number 25 kilohertz or less. The value selected here, 2.20 kilohertz, is chosen to be uncorrelated with any other frequencies present in the system.

While the chip period of the desired signal a priori discriminant function is assumed to be arbitrary, we do require that it be an integer. This simplifies the programming without substantially reducing generality of the program. In this example, a very slow discriminant function is being simulated since the period is 660 microseconds (1515 hertz). A discriminant function varying this slowly represents a moderately difficult case for the adaptive array in that the time required to decorrelate

the desired signal and jammer is inversely proportional to this frequency.

Real adaptive array weights and real microprocessors have limited dynamic range. We include these effects by specifying maximum ranges on the array weights, the alpha loop weight and the receiver AGC. Respectively, these parameters are AR, AH, GC. In the simulation, the various weights are hard-limited at the specified levels.

This simulation program can be run in a number of different modes. Twenty parameters identify these modes although only a few parameters are frequently changed. Note that these options are divided into four groups of five each. They are numbered consecutively from left to right. We will identify only those parameters which are of significance here.

TABLE 5.3-1. OPTION PARAMETER NUMBER IDENTIFICATION

<u>Option Number</u>	<u>Effect</u>
10	0 Update weights continually
	1 Update weights at the end of a perturbation period
15	0 Normal
	1 Disconnect signal recognizer
18	0 Normal
	1 Flag weight limiting
19	0 Max precision
	2 Truncate D/A precision (QR)
	3 Truncate weight precision (QW)
	1 Truncate both the QR and QW

Next, an identifying description of the type of antenna used is given. In this case, the words spiral, 0.5, refers to a quasi logarithmically placed array whose first element is located at $.5\lambda$ from the center. It is necessary to refer to a table to obtain spacing of all of the elements. However, this is the only antenna used in the simulations described in this section. The element spacings are:

TABLE 5.3-2. ANTENNA ELEMENT SPATIAL COORDINATES

<u>Element No.</u>	<u>X</u>	<u>Y</u>	<u>Z</u>
1	0.5	0.0	0.0
2	0.0	0.57	0.0
3	-0.61	0.0	0.0
4	0.0	-0.9	0.0

Although it is not given on this standard case output, an additional parameter appears under the antenna section when the array is rotated. This parameter expresses the rotation rate of the antenna in degrees per second about each of the X, Y and Z axes, respectively. These rate parameters are KX, KY, KZ.

The last of the parameters are those forming the emitter descriptions. The first emitter described is always the desired signal. Each emitter is described by first giving its power in dB with respect to thermal noise at a reference given antenna element input. Three additional parameters follow. They are X, Y and Z distances to the emitter expressed in arbitrary units. Consequently, these numbers describe angles of arrival in Cartesian coordinates. In this standard case, the signal power is 20 dB with respect to thermal noise. The signal is located at +10 units along the Y axis. Since both X and Z coordinates are zero, the signal angle of arrival is 90° azimuth, zero

elevation. A single jammer is present, the power of which is 40 dB with respect to thermal noise. This jammer also has a zero Z coordinate but has an X coordinate of 0.87 (actually 0.866) and a Y coordinate of 0.5, thus an angle of arrival of 30° with respect to the X axis.

Let us now examine Figure 5.3-1 so as to determine how the algorithm performed. Input signal-to-noise ratio is -20 dB. Note that output signal power with respect to thermal noise begins at 20 dB and immediately rises to about 21 dB. This initial rise is due to two factors. Initial gain enhancement is due to antenna pattern lobes accompanying the jammer null. The slow rise in signal output power is due to application of the desired signal steering vector by the algorithm's alpha loop.

Suppression of the jammer as shown here is anomalous in that a straight line power reduction is usually seen. In this example, the array is allowed to update its weights on a continual basis, that is, at the end of each individual weight perturbation rather than at the end of a perturbational period (option 10=0). While the reason is not clear, allowing such updating has almost invariably produced faster than expected initial adaptation transients followed by slower than expected transients. "Steady-state" values thereafter are practically identical to those of the periodic update case.

Let us now calculate a theoretical adaptation rate for this configuration and compare it with the simulation results. Using equations from Section 3.1.1, we can show that the jammer suppression will be described as equation having the following form:

$$\frac{P(t)}{P(0)} = e^{-2K\lambda_j T} \quad 5-8$$

where the term K is algorithm loop gain and λ_j is the jammer eigenvalue. The factor of 2 is obtained from squaring W in obtaining output power. A good approximation to the jammer eigenvalue is obtained by multiplying

jammer power times the number of weighted antenna inputs. (This assumes that the steering vector for the jammer and the desired signal are not approximately colinear.) For this case of a four-element array having one element fixed weighted, we get

$$\lambda_j \approx (4-1)(2)(10^4) = 6 \times 10^4 \quad 5-9$$

where we have recognized that each weight consists of a complex pair. A power value of 10^4 is used because the algorithm requires input parameters normalized with respect to thermal noise. On an average basis, the algorithm has obtained a 40 dB jammer suppression in about 18 milliseconds. Thus, we may write:

$$10[-2K\lambda_j T] \log(e) = 10 \log \frac{P(18 \text{ ms})}{P(0)} = -40 . \quad 5-10$$

The parameter T is not real time but is instead the number of iterations. This is due to the fact that we are evaluating a difference equation instead of a differential equation, therefore:

$$T = \frac{18 \text{ ms}}{20 \mu\text{s}} = 900 . \quad 5-11$$

Solving for loop gain K yields:

$$K = \frac{-4.0}{-2\lambda_j T \log e} \approx 8.53 \times 10^{-8} . \quad 5-12$$

Alternatively, the total loop gain may be calculated from knowledge of K_A , K_P and a parameter K_W , not mentioned previously, which is due to the

method used by the microprocessor to represent weights. Specifically, the weights are assumed to be represented by a fixed point integer, one bit of which is reserved for sign. In effect, loss of the sign bit in representing weight magnitude introduces a factor of one-half in the AMD 2900 algorithm. We get:

$$K' = 2(KA)^2(KP)(KW)(32/33) \approx 9.70 \times 10^{-8} \quad 5-13$$

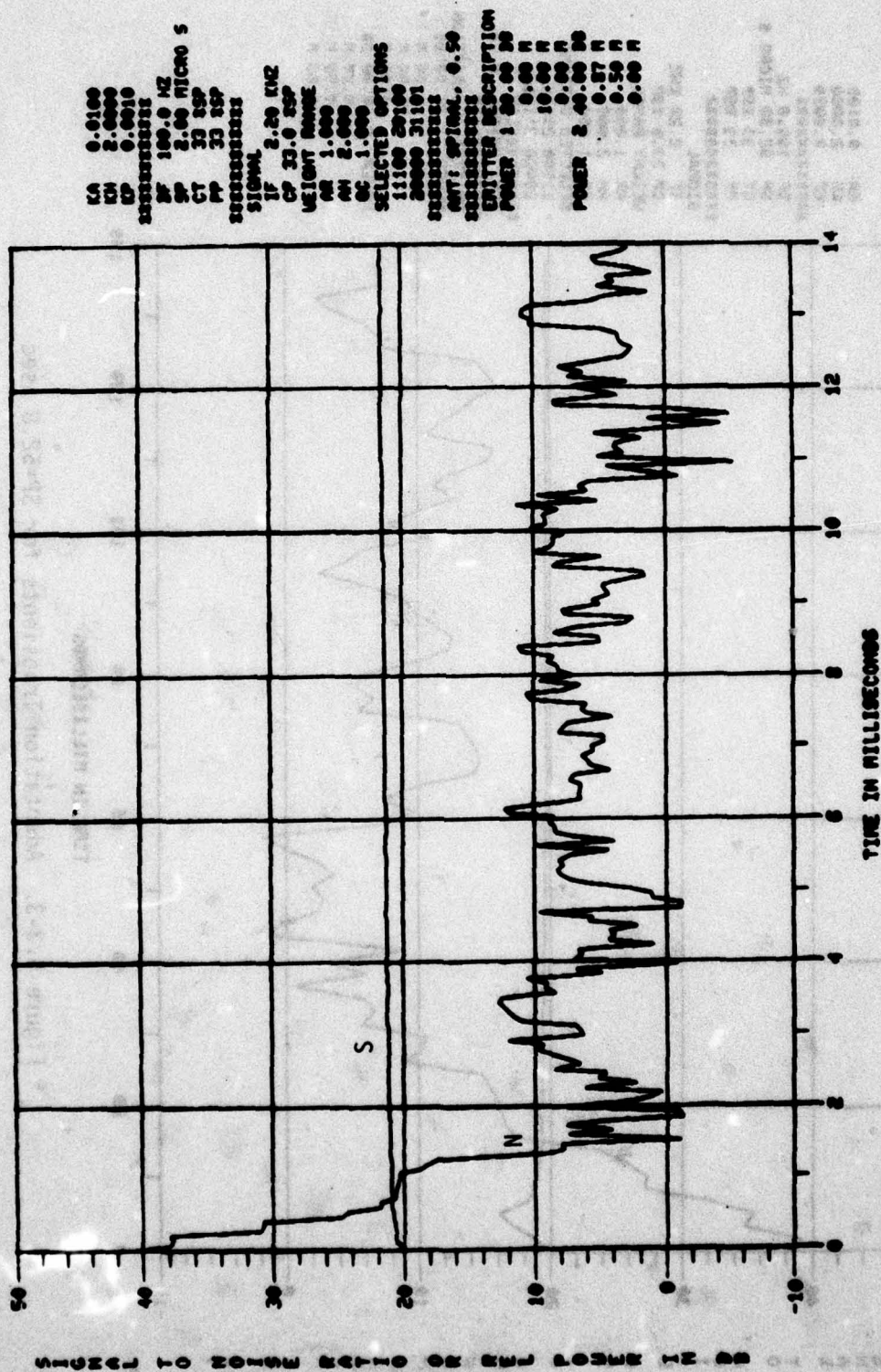
where the factor 32/33 arises from the fact that the 33rd term in the weight perturbation is zero. The factor of 2 in (5-13) is due to the fact that the perturbational term in the squaring multiplier outputs has a coefficient of 2. Comparison of K and K' gives:

$$K \approx (0.88)K' \quad 5-14$$

This is reasonably good agreement considering that we are evaluating a stochastic simulation with continual adaptive update.

An adaptation curve for SP=2 microseconds is given in Figure 5.3-2. Except for the fact that the time scale is different by an order of magnitude, as expected, this curve is basically similar to the 20 microsecond case. Observe that the jammer is suppressed to a few dB above the output thermal noise level (zero dB on the graph). Since we are plotting total noise power, one only expects occasional small deviations of the plotted value below the 0 dB point, due to instantaneously small values of output thermal noise.

Figure 5.3-3 illustrates an adaptation transient obtainable by the actual AMD 2900 microprocessor circuitry. The nulling speed change is evident. Also, noise power in this example is not suppressed as much as in the previous cases. This is thought due to the fact that the a priori discriminant is becoming very slow and significant cross correlation between the desired signal, the IF carrier frequency and the



KA 0.0100
 KB 2.0000
 KC 0.0010

 SF 100.0 KHZ
 SP 2.00 MICRO S
 CT 33 KSP
 PT 33 KSP

 SIGNAL
 IF 2.20 KHZ
 CF 33.0 KSP
 WEIGHT RANGE
 AR 1.000
 AM 2.000
 AC 1.000

 SELECTED OPTIONS
 1100 20100
 20000 31101

 ANT: OPTICAL, 0.50

 Emitter Description
 POWER 1 20.00 DB
 0.00 H
 10.00 H
 0.00 H
 0.00 H
 POWER 2 40.00 DB
 0.00 H
 0.00 H
 0.00 H
 0.00 H

Figure 5.3-2. Adaptation Transients for SP=2 μ sec

UNOZED TO SOME EXTENT ON THE LOWER 12 IN

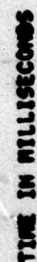


Figure 5.3-3. Adantation Transients for SP=52.8 μ sec

Walsh perturbations is occurring. Furthermore note that the desired signal is not being enhanced as much as in the previous examples. This performance is typical of excessive signal/jammer and signal/IF frequency correlations. An adjustment of CP and IF would yield better results. *

5.3.2 Array Gain

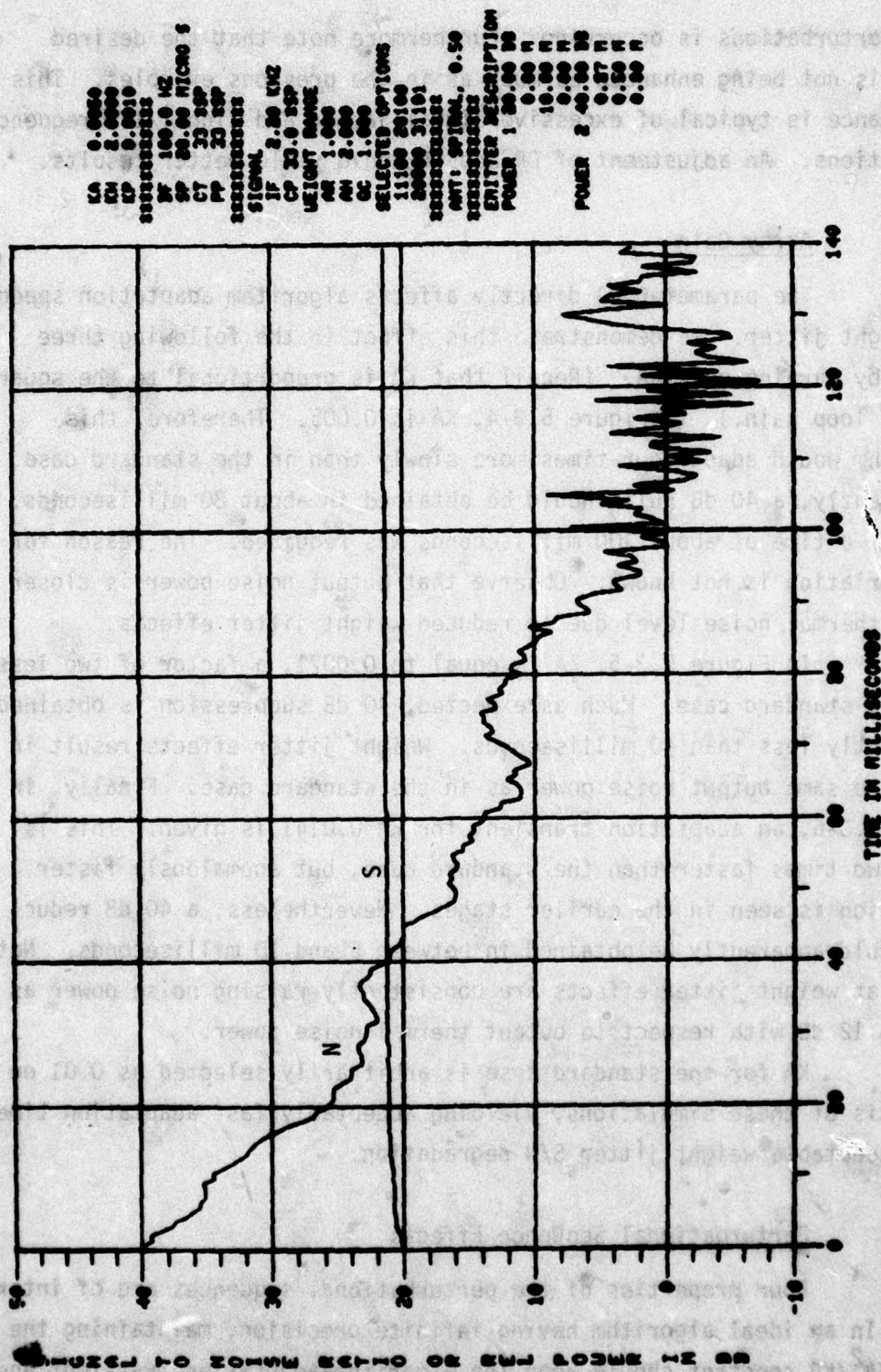
The parameter KA directly affects algorithm adaptation speed and weight jitter. We demonstrate this effect in the following three graphs by varying only KA. (Recall that KA is proportional to the square root of loop gain.) In Figure 5.3-4, KA is 0.005. Therefore, this algorithm would adapt four times more slowly than in the standard case. Consequently, a 40 dB null should be obtained in about 80 milliseconds. Instead, a time of about 100 milliseconds was required. The reason for this variation is not known. Observe that output noise power is closer to the thermal noise level due to reduced weight jitter effects.

In Figure 5.3-5, KA is equal to 0.0071, a factor of two less than the standard case. Much as expected, 40 dB suppression is obtained in slightly less than 40 milliseconds. Weight jitter effects result in about the same output noise power as in the standard case. Finally, in Figure 5.3-6, an adaptation transient for KA=0.0141 is given. This is about two times faster than the standard case, but anomalously faster adaptation is seen in the earlier stages. Nevertheless, a 40 dB reduction would apparently be obtained in between 5 and 10 milliseconds. Note here that weight jitter effects are consistently raising noise power as much as 12 dB with respect to output thermal noise power.

KA for the standard case is arbitrarily selected as 0.01 on the basis of these simulations, yielding acceptably fast adaptation times with acceptable weight jitter S/N degradation.

5.3.3 Perturbational Sequence Effects

Four properties of the perturbational sequences are of interest here. In an ideal algorithm having infinite precision, maintaining the ratio KA^2/KP constant should keep the algorithm adaptation transient and



LA 0.0000
 KA 0.0000
 CP 0.0010
 SF 100.0 HZ
 SP 20.00 MICRO S
 CT 23.0 SP
 PP 23.0 SP
 SIGNAL
 IF 2.00 KHZ
 CP 23.0 SP
 WEIGHT NAME
 AN 1.000
 AN 2.000
 AC 1.000
 SELECTED OPTIONS
 11100 20100
 20000 31101
 ANT: SPINAL, 0.50
 ENITTER DESCRIPTION
 POWER 1 20.00 W
 0.00 H
 10.00 H
 0.00 H
 0.57 H
 0.50 H
 0.00 H
 POWER 2 40.00 W
 0.00 H
 0.57 H
 0.50 H
 0.00 H

Figure 5.3-4. Adaptation Transients for KA=0.005

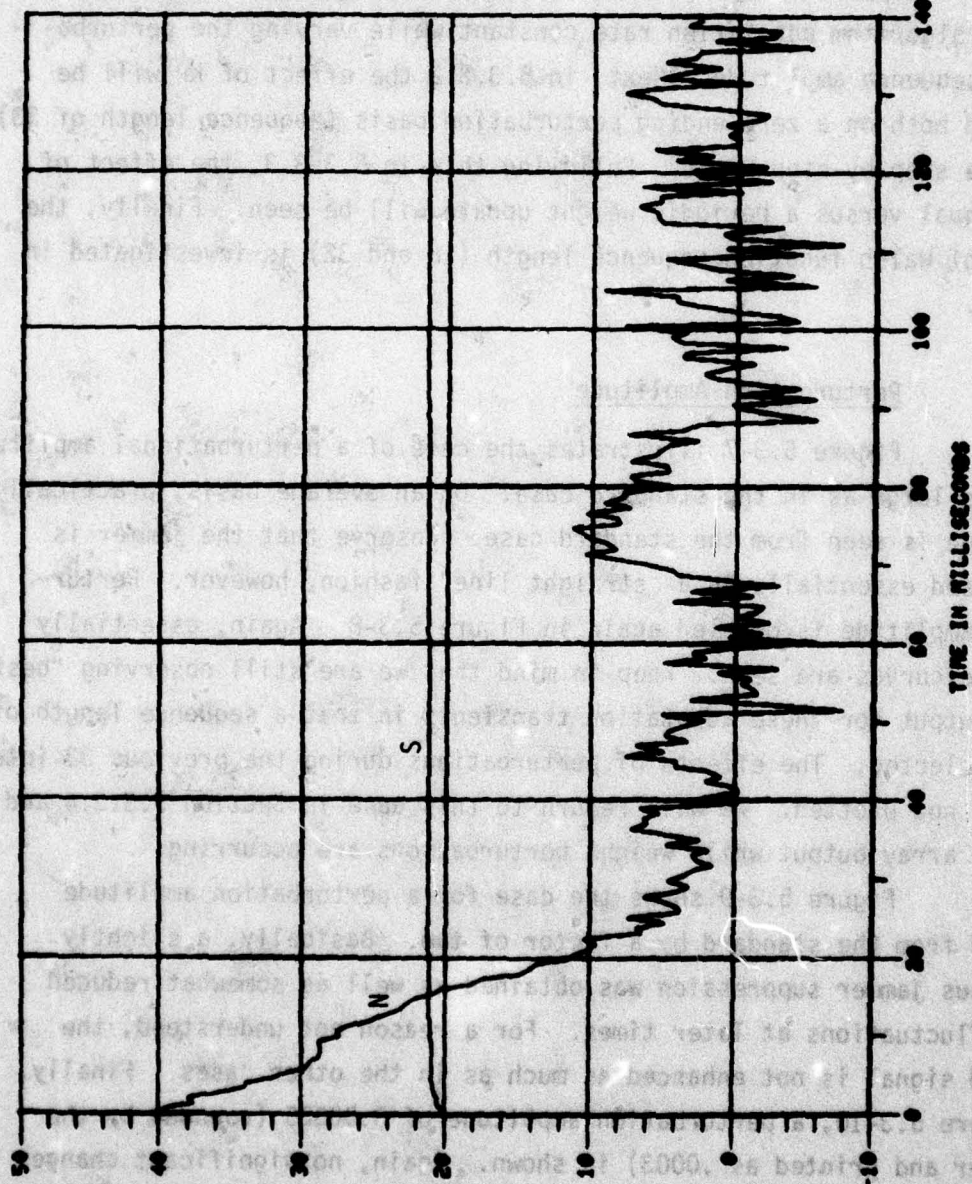


Figure 5.3-5. Adaptation Transients for KA=0.00707

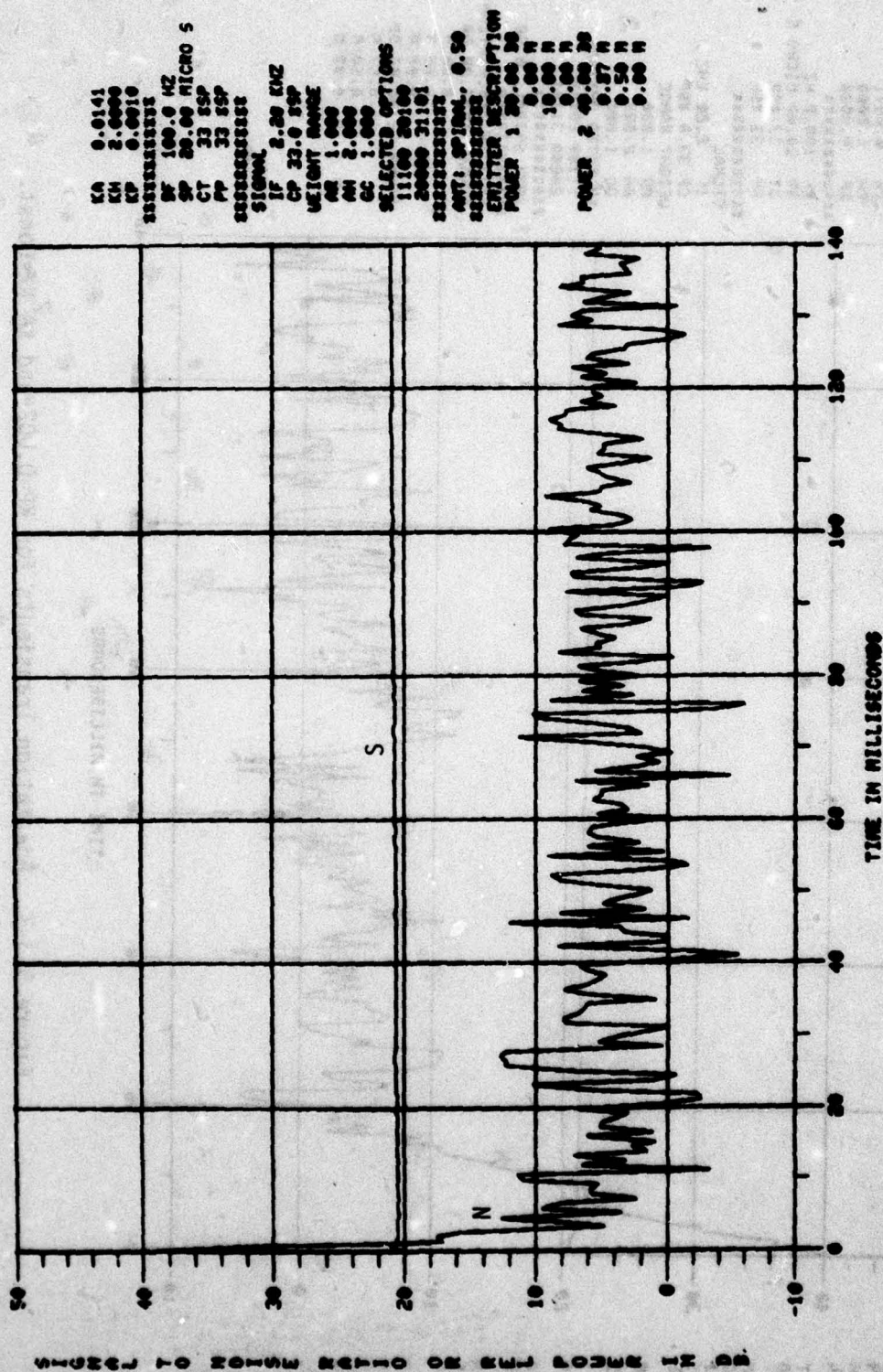
weight jitter effects constant. Due to finite precision, KP too small will result in an erroneous gradient being measured and zero or erratic adaptation will be obtained. Alternatively, KP too large will cause excessive S/N degradation due to the weights being substantially perturbed from their optimum values.

Specifically, we will first, in 5.3.3.1, be concerned with keeping algorithm adaptation rate constant while varying the perturbational sequence amplitude. Next, in 5.3.3.2 the effect of KP will be observed both on a zero ending perturbation basis (sequence length of 33) and on a step-by-step basis. Following this in 5.3.3.3, the effect of a continual versus a periodic weight update will be seen. Finally, the effect of Walsh function sequence length (16 and 32) is investigated in 5.3.3.4.

5.3.3.1 Perturbation Amplitude

Figure 5.3-7 illustrates the case of a perturbational amplitude twice as large as in the standard case. On an average basis, practically no change is seen from the standard case. Observe that the jammer is suppressed essentially in a "straight line" fashion, however. Perturbation amplitude is doubled again in Figure 5.3-8. Again, essentially the same curves are seen. Keep in mind that we are still observing "best case" output for these adaptation transients in that a sequence length of 33 is selected. The effects of perturbations during the previous 32 intervals is not plotted. We will return to this case in Section 5.3.3.4 and observe array output while weight perturbations are occurring.

Figure 5.3-9 shows the case for a perturbation amplitude reduced from the standard by a factor of two. Basically, a slightly anomalous jammer suppression was obtained as well as somewhat reduced noise fluctuations at later times. For a reason not understood, the desired signal is not enhanced as much as in the other cases. Finally, in Figure 5.3-10, a perturbation amplitude of 0.00025 (rounded by the computer and printed as .0003) is shown. Again, no significant changes are seen in the jammer suppression curve from the previous case, but the desired signal gain was relatively increased about 1 dB.



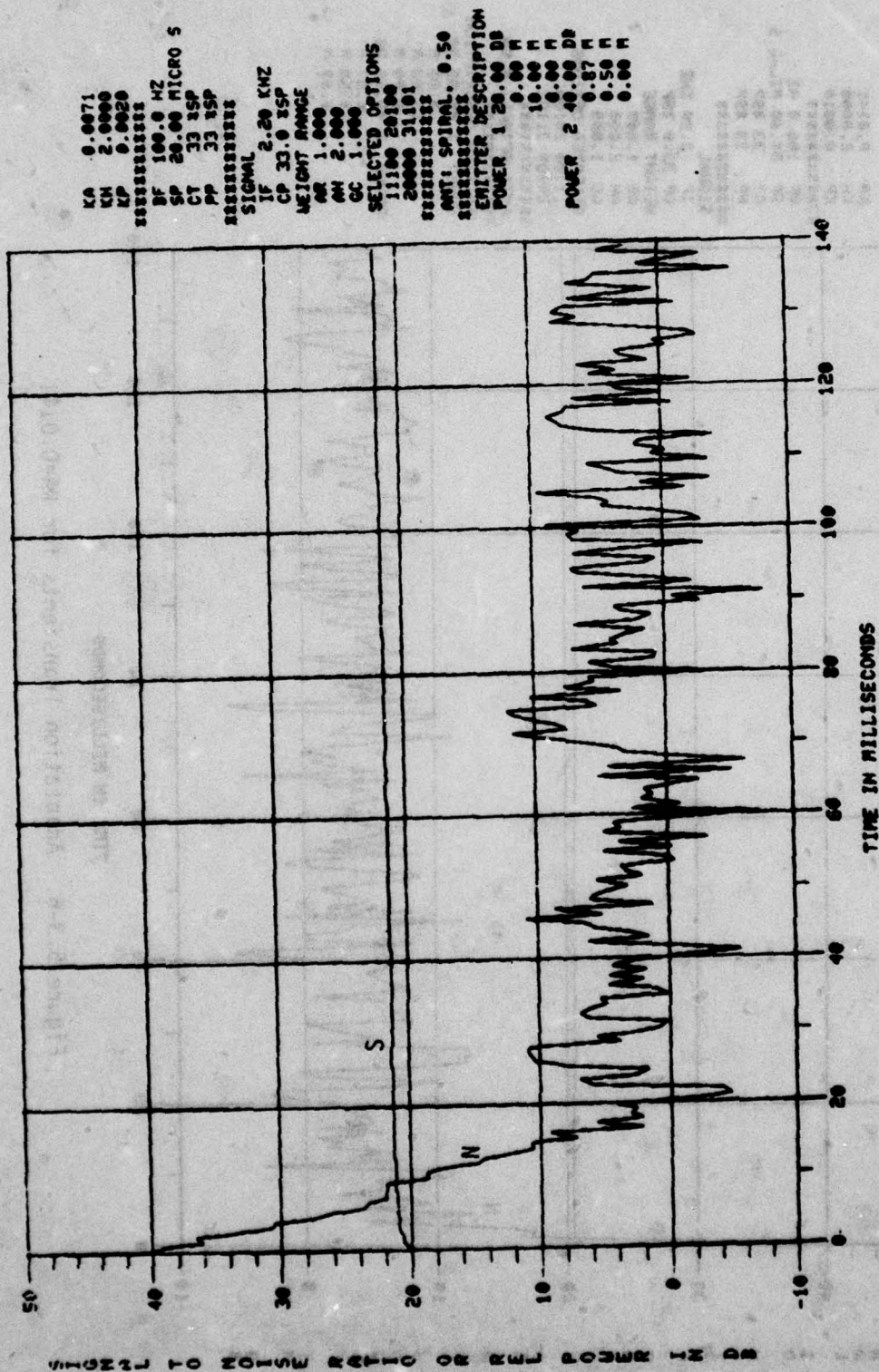


Figure 5.3-7. Adaptation Transients for $KP=0.002$ and $KA^2 KP=Const.$

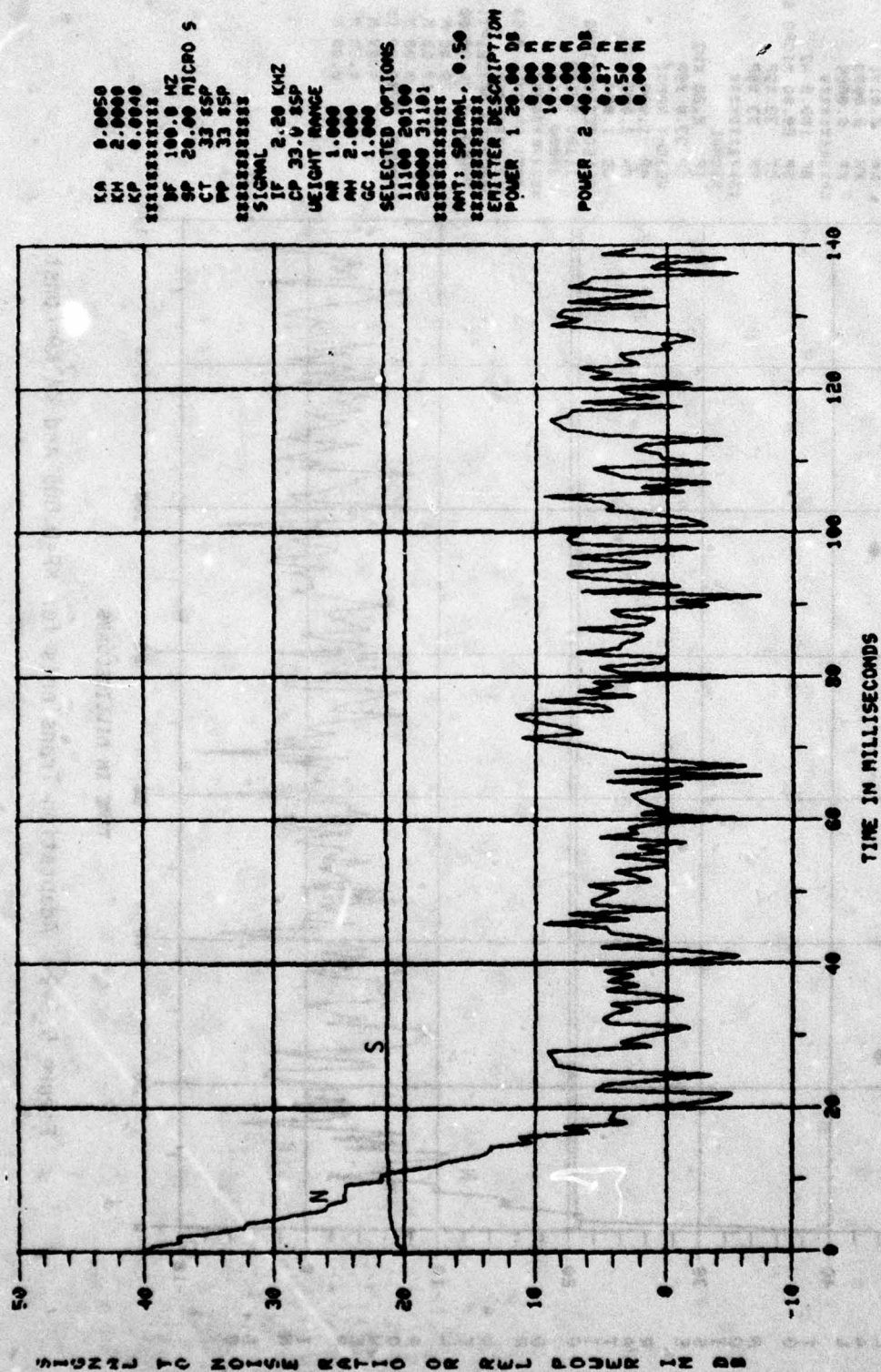


Figure 5.3-8. Adaptation Transients for $KP=0.004$ and $KA^2 KP=Const.$

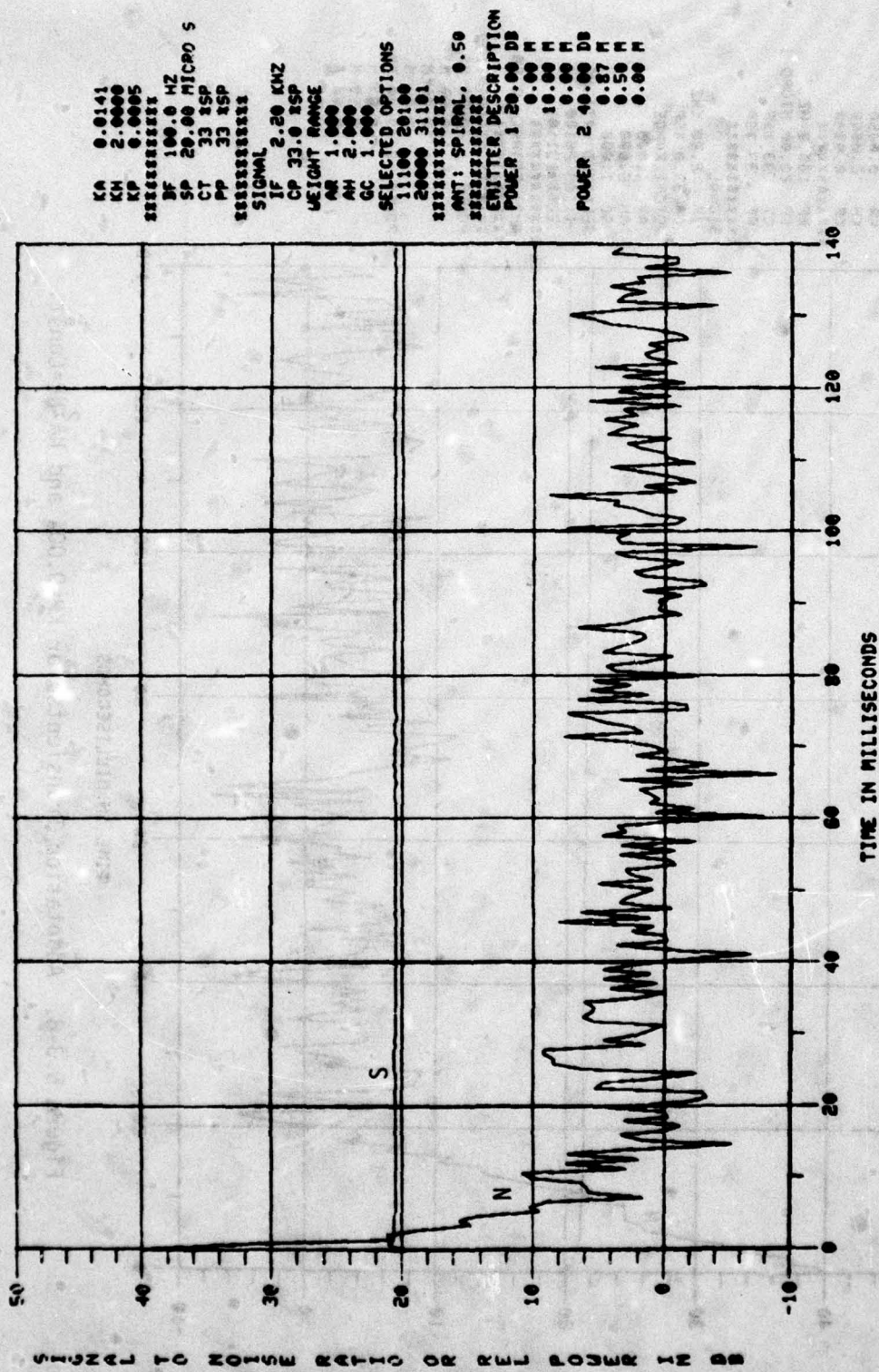


Figure 5.3-9. Adaptation Transients for $KP=0.005$ and $KA^2KP=Const.$

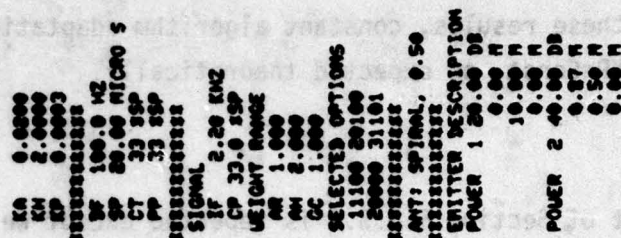


Figure 5.3-10. Adaptation Transients for $K_P=0.00025$ and $K_A^2 K_P=\text{Const.}$

According to these results, constant algorithm adaptation rate is obtained if $(KA)^2 KP = \text{Const}$, as expected theoretically.

5.3.3.2 Step-by-Step

The experiment of Section 5.3.3.1 is repeated except we evaluate array output at each perturbational step rather than at the end of a cycle of 33. In the examples given here the array is allowed to update its weights only at the end of a cycle of 33 perturbations rather than on a continual basis. Figure 5.3-11 is for the standard case. Since $PP=33$, 660 microseconds is required per weight update. Weight update steps are clearly shown in the figure. Also, the effect of the individual fluctuations can be clearly seen. Observe that signal power is almost unaffected (due to the fact that a pattern lobe exists toward the desired signal) while jammer becomes progressively more affected as the null is approached. The parameter of most interest here is not peak-to-peak jammer variation due to the perturbations, but the amplitude of the perturbation with respect to the expected value of the jammer power. Here, $KP=0.001$ results in about a 2 dB degradation worst case and much less than that on the average.

Figure 5.3-12 is for a weight perturbation two and one-half times greater than previously. Although greater variation is seen in jammer power, expected value degradations are still not excessively large.

Finally in 5.3-13, a 0.004 perturbation causes 4 to 6 dB degradation peak in jammer null loss at times during the cycle. Still, average value degradation of jammer output power is probably less than 1 dB.

5.3.3.3 Continual/Periodic Weight Update

In these paragraphs, we will examine the effect of continual versus periodic weight updating. Recall that in the previous example, the weights were updated only at the end of a perturbational sequence. According to the derivations given in Chapter 3.0, particularly those relating to the expected value equivalence of a random search algorithm,

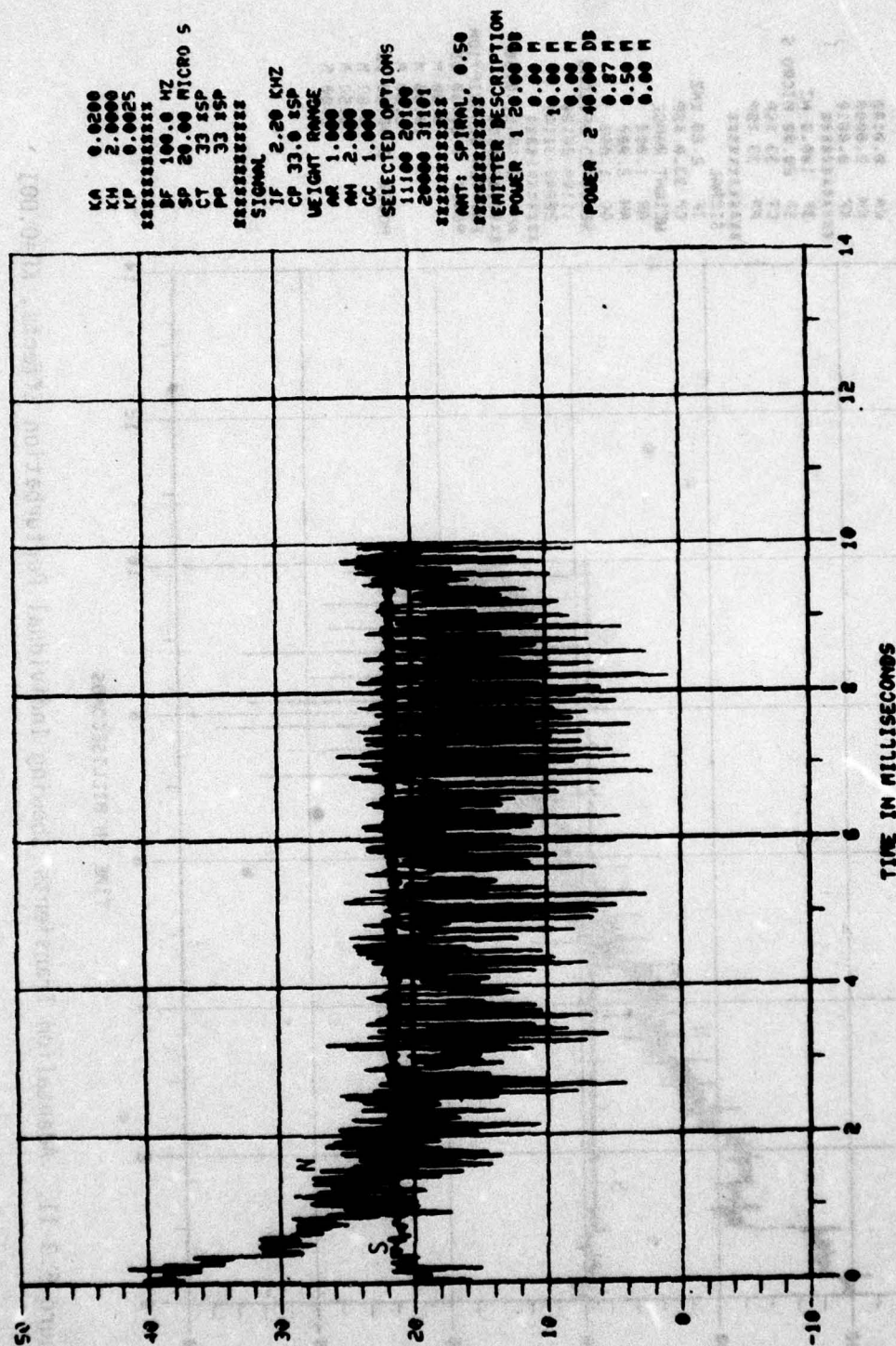


Figure 5.3-12. Adaptation Transients Showing Individual Perturbation Effects, KP=0.0025

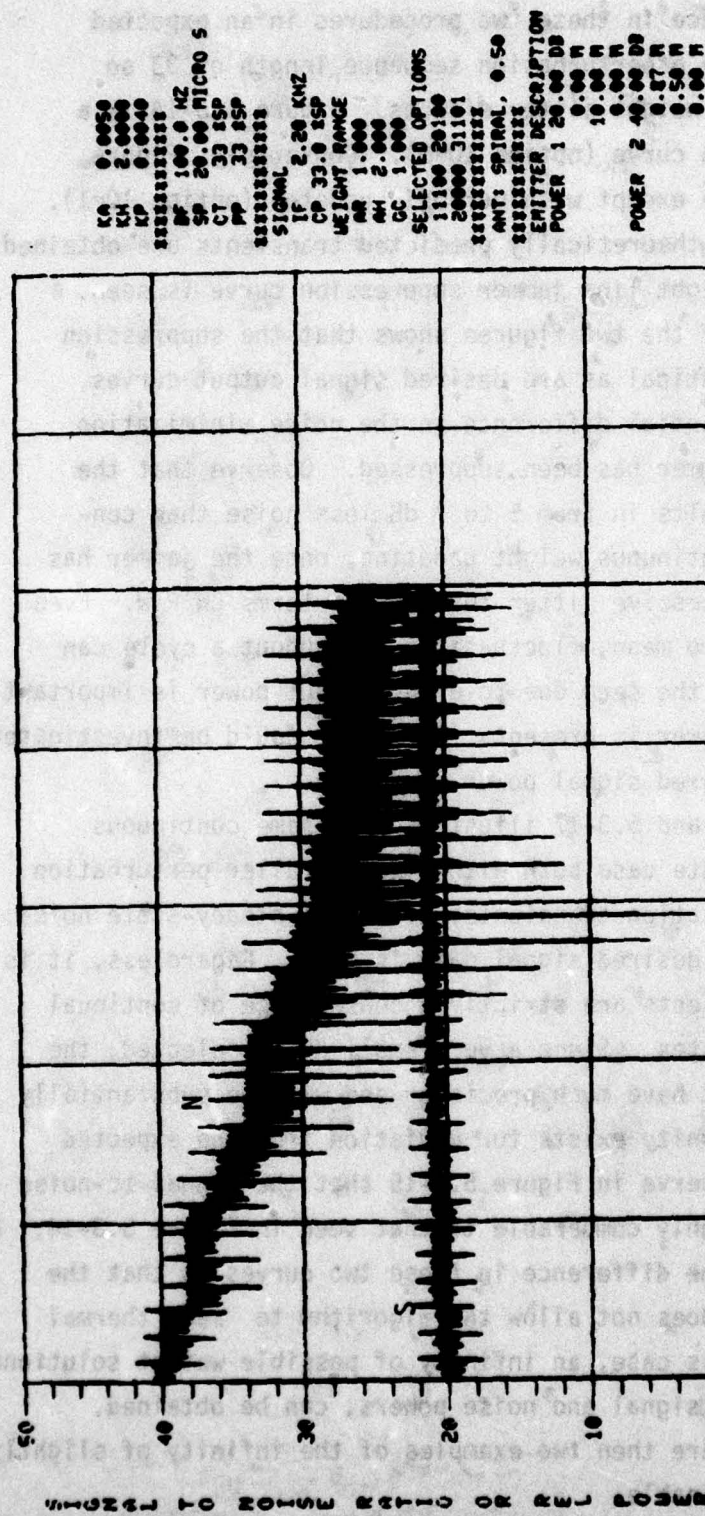


Figure 5.3-13. Adaptation Transients Showing Individual Perturbation Effects, KP=0.004

there should be no difference in these two procedures in an expected value sense. Again, we use a perturbation sequence length of 33 so as to enable separation of weight jitter effects. Figure 5.3-14 is a continual update adaptation curve (option 10=0). Conversely, Figure 5.3-15 is for the same case except with periodic updates (option 10=1).

In this case, theoretically predicted transients are obtained. Note that an expected straight line jammer suppression curve is seen. Furthermore, examination of the two figures shows that the suppression curves are practically identical as are desired signal output curves. There is, however, a substantial difference in the noise minimization achieved well after the jammer has been suppressed. Observe that the periodic weight update results in from 5 to 7 dB less noise than continuous weight update. Continuous weight updating, once the jammer has been nulled, introduces excessive jitter through the terms $CW^T R_x W$. Even though these terms have zero mean, fluctuations throughout a cycle can be substantial. Note that the term due to array output power is important since substantial signal power is present. This case could be investigated further by varying the desired signal power parameter.

Figure 5.3-16 and 5.3-17 illustrate the same continuous versus periodic weight update case both with a much smaller perturbation parameter. Different adaptation transients, different steady-state noise minimization and different desired signal gain is seen. Regardless, it is not believed that these effects are strictly a consequence of continual versus periodic weight updates. Since a very small KP is selected, the estimated gradient will not have much precision and will be substantially noisy. Thus, ample opportunity exists for deviation from the expected gradient. Furthermore, observe in Figure 5.3-15 that the signal-to-noise ratio obtained is only roughly comparable to that seen in Figure 5.3-14. A plausible explanation of the difference in these two curves is that the small weight perturbation does not allow the algorithm to "see" thermal noise correlations. In this case, an infinity of possible weight solutions, each with differing output signal and noise powers, can be obtained. Figures 5.3-14 and 5.3-15 are then two examples of the infinity of slightly non-optimum solutions obtainable.

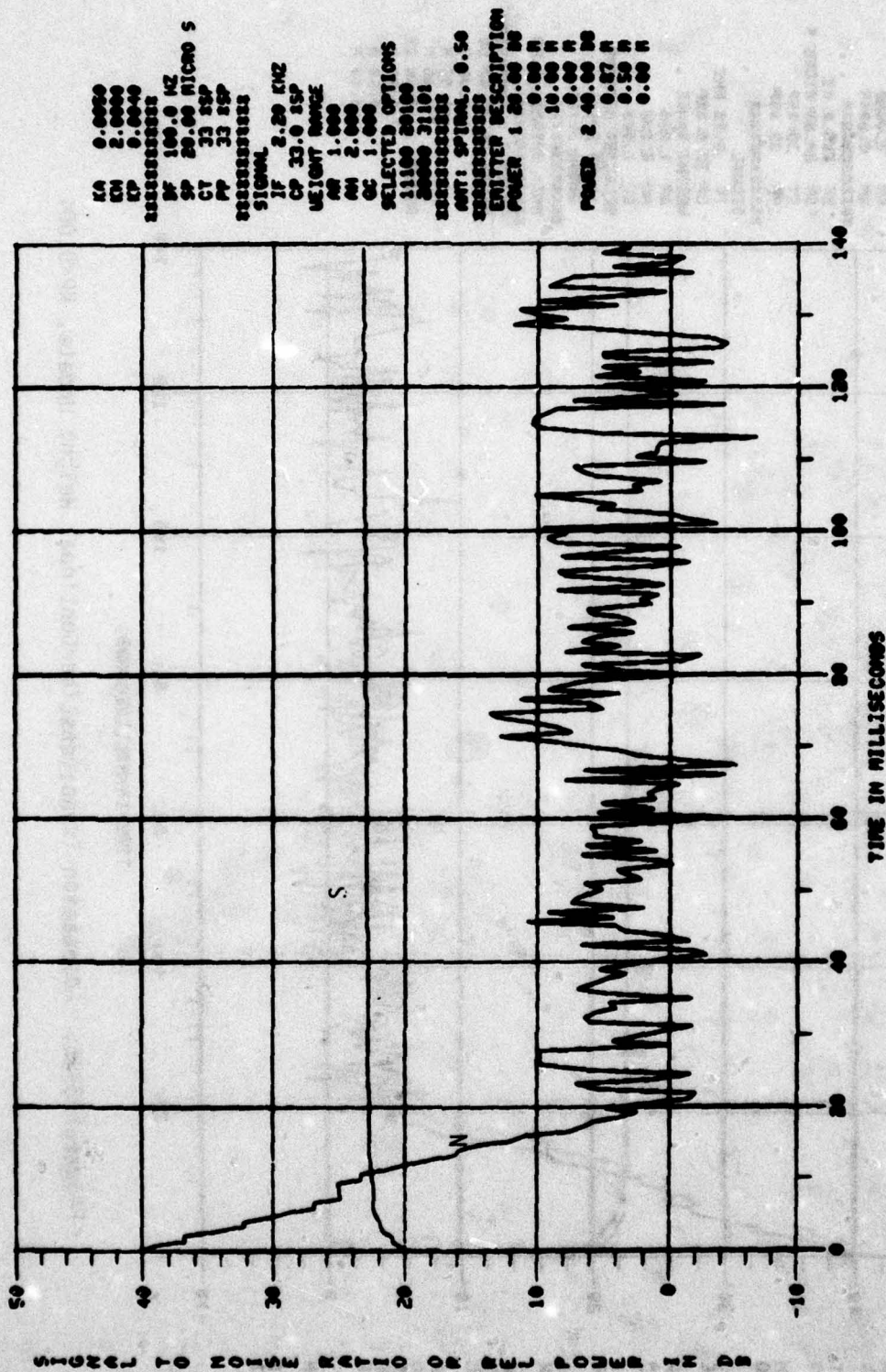
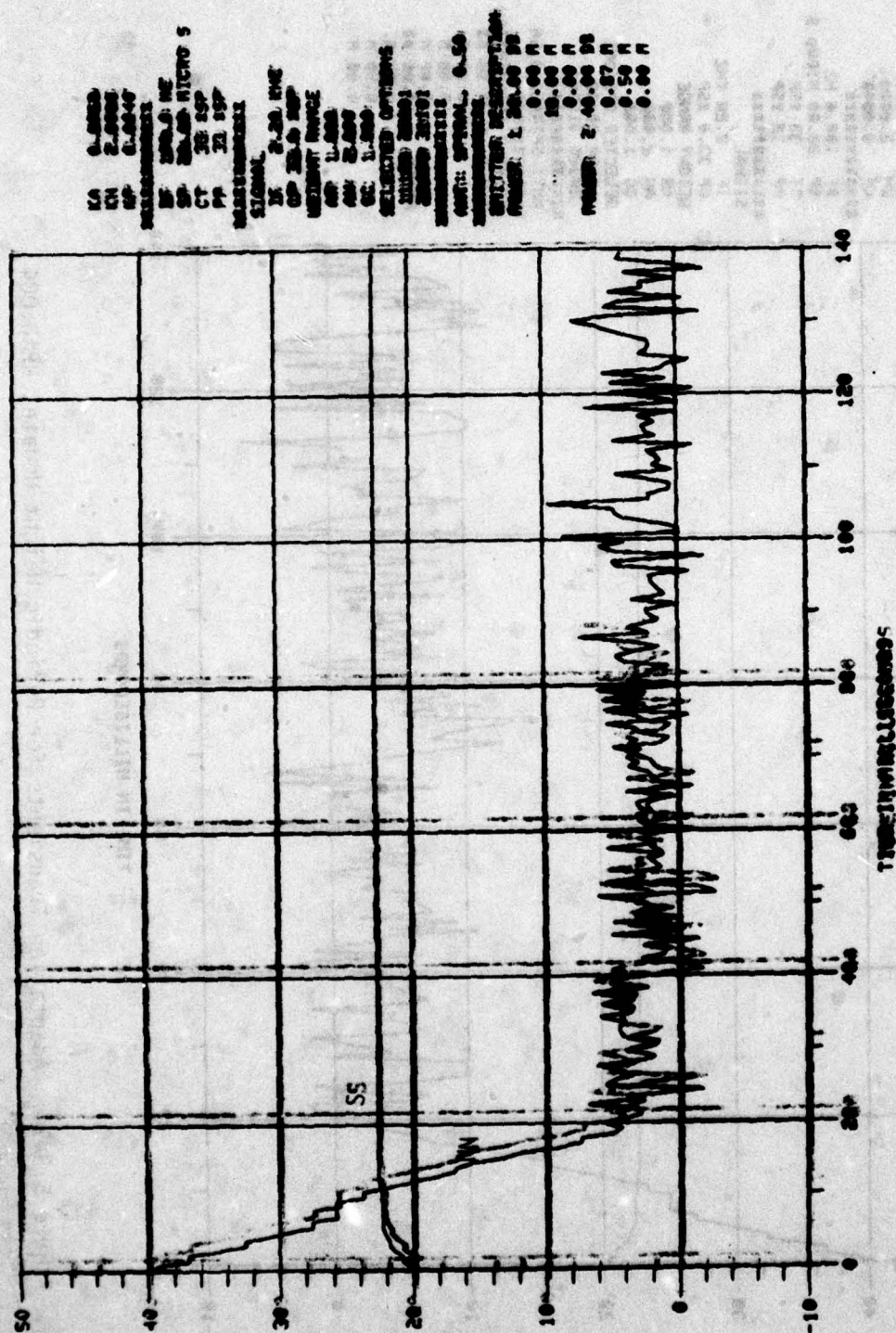


Figure 5.3-14. Adaptation Transients for Periodic Weight Update, $KP=0.004$



EA 0.0000
EM 2.0000
EP 0.0003

RF 100.0 KHZ
SP 20.00 MICRO S
CT 33 SPP
PP 33 SPP

SIGNAL 2.30 KHZ
IF 33.0 SPP
WEIGHT RANGE
AR 1.000
AM 2.000
SC 1.000

SELECTED OPTIONS
11100 20100
20000 21101

AWT: SPINNA. 0.50

SWITCH DESCRIPTION
POWER 1 20.00 DB
0.00 H
10.00 H
0.00 H
POWER 2 40.00 DB
0.57 H
0.50 H
0.00 H

TIME IN MILLISECONDS

Figure 5.3-16. Adaptation Transients for Continual Weight Update, $KP=0.0003$

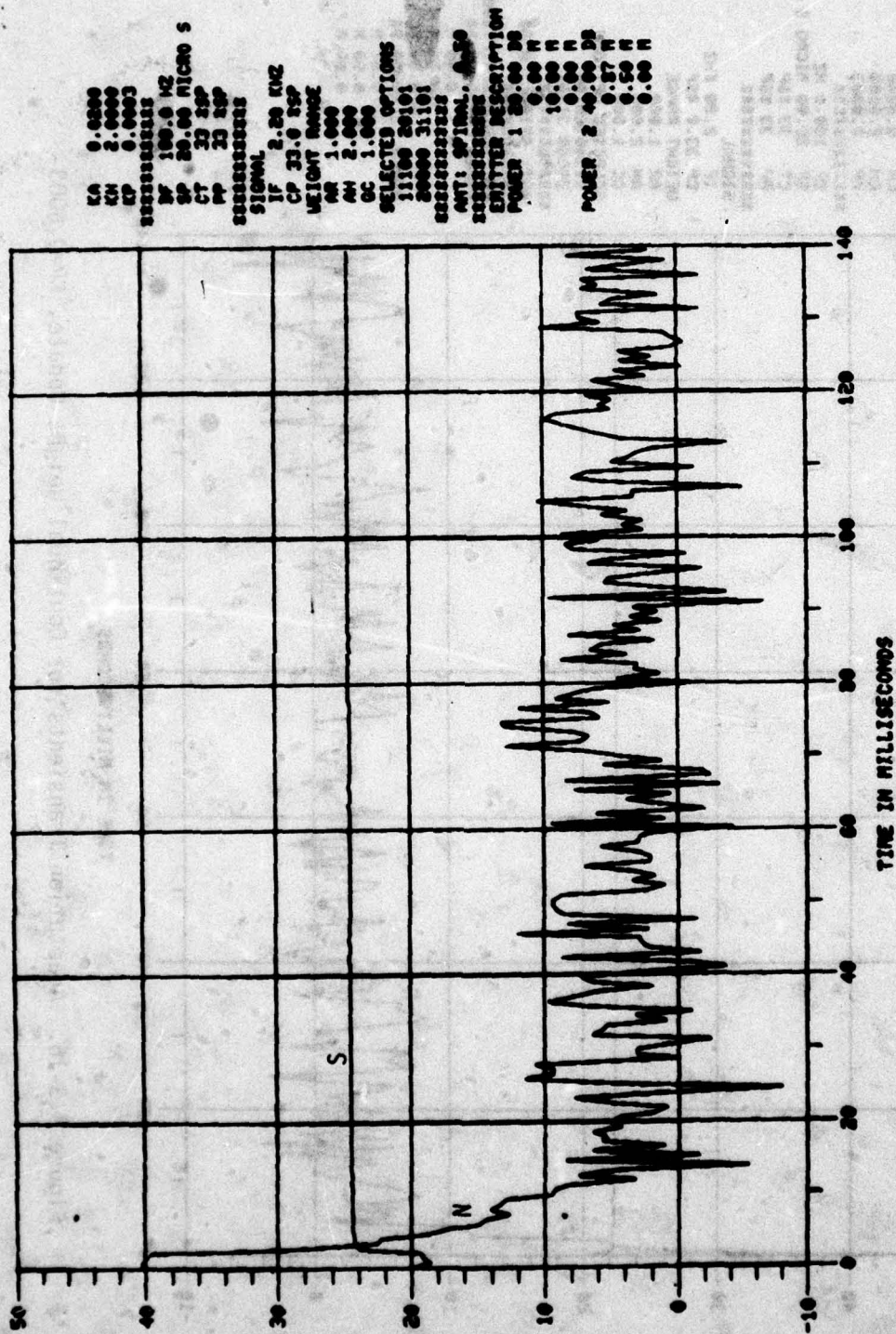


Figure 5.3-17. Adaptation Transients for Periodic Weight Update, KP=0.0003

In the previous four cases, the array was rotated through an angle of 45° with respect to the standard condition in order to rule out angle of arrival effects as a cause of the anomalous adaptation transients for small KP. Such cases may then be compared with Figure 5.3-10 for the same amount of KP. Since a similar adaptation transient is obtained, the effect is probably not a function of antenna orientation or signal and jammer angles of arrival.

5.3.3.4 Perturbation Sequence Length

Algorithm adaptation is investigated for sequence lengths of 16 and 32 terms. All updating is done periodically at the end of a perturbational sequence. Except for variations of KP and KA so as to maintain a constant array adaptation gain, the standard case is run. Figure 5.3-18 (for 32) and Figure 5.3-19 (for 16) are seen to be essentially identical. (Exact agreement would not be expected since updating occurs at different times with respect to the IF carrier frequency term as well as at different times with respect to the desired signal pseudo random sequence.) Observe that a somewhat higher output noise power is seen than in the perturbation length of 33 units. In these curves, we are seeing the affect of the individual weight perturbation terms illustrated earlier in Section 5.3.3.2. Though these curves may not be representative of average noise power, being somewhat pessimistic in their predictions, they nevertheless do indicate that degradation due to KP is occurring.

We may similarly compare Figures 5.3-20 and 5.3-21 which are run for the case of $KP=0.004$. Again, output signal and noise power curves are practically identical. Note that substantial noise minimization degradation has occurred in this case. Since KP is approximately four times this standard value, jitter degradation would be roughly 16 times as bad, or roughly a degradation of 12 dB. Comparison of 5.3-19 and 5.3-21 shows this to be almost precisely the degradation obtained.

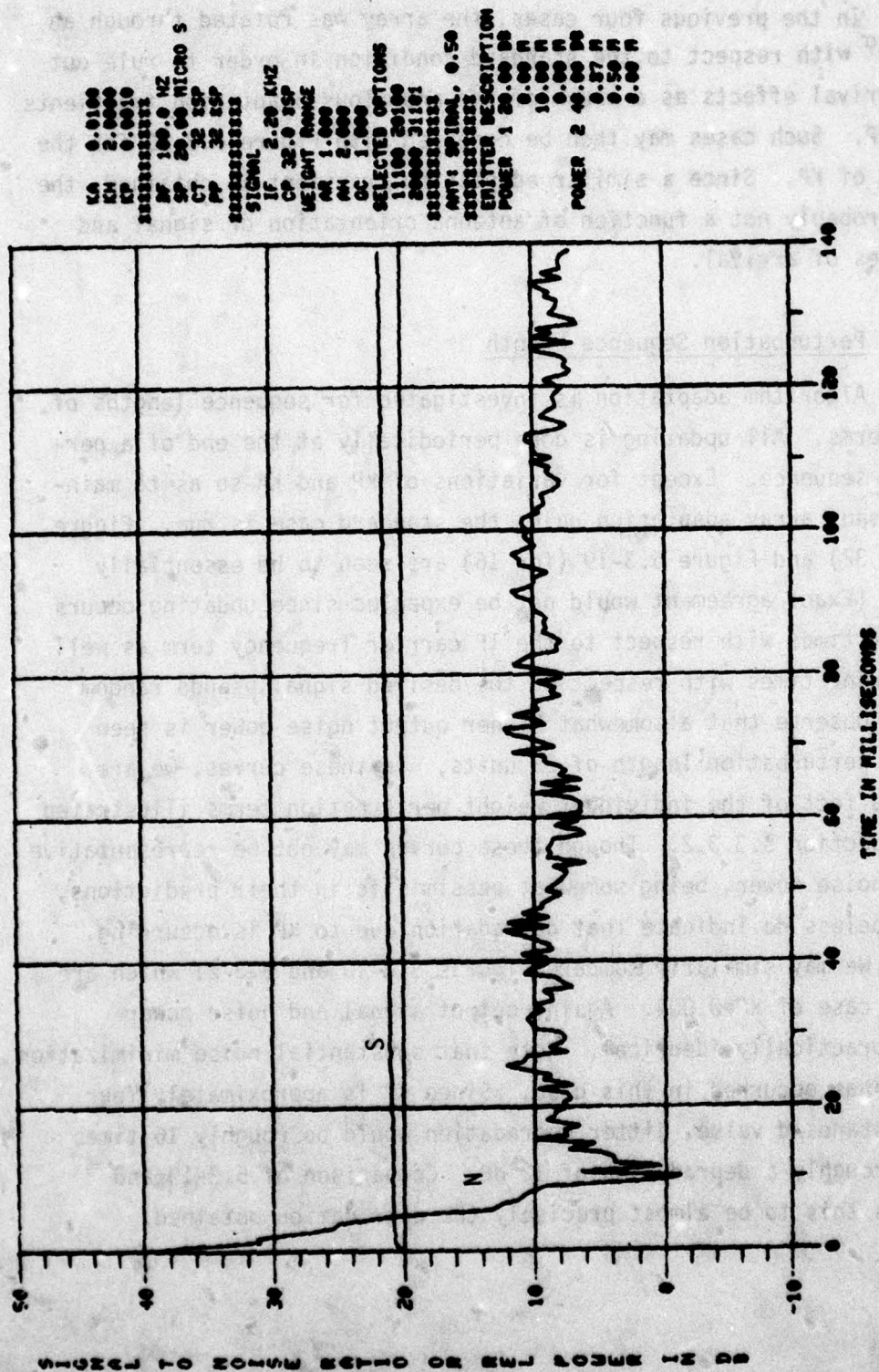
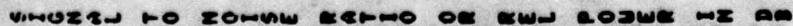


Figure 5.3-18. Adaptation Transients for a Perturbation Sequence Length of 32 and KP=0.001



Figure 5.3-19. Adaptation Transients for a Perturbation Sequence Length of 16 and $K_P=0.001$



222

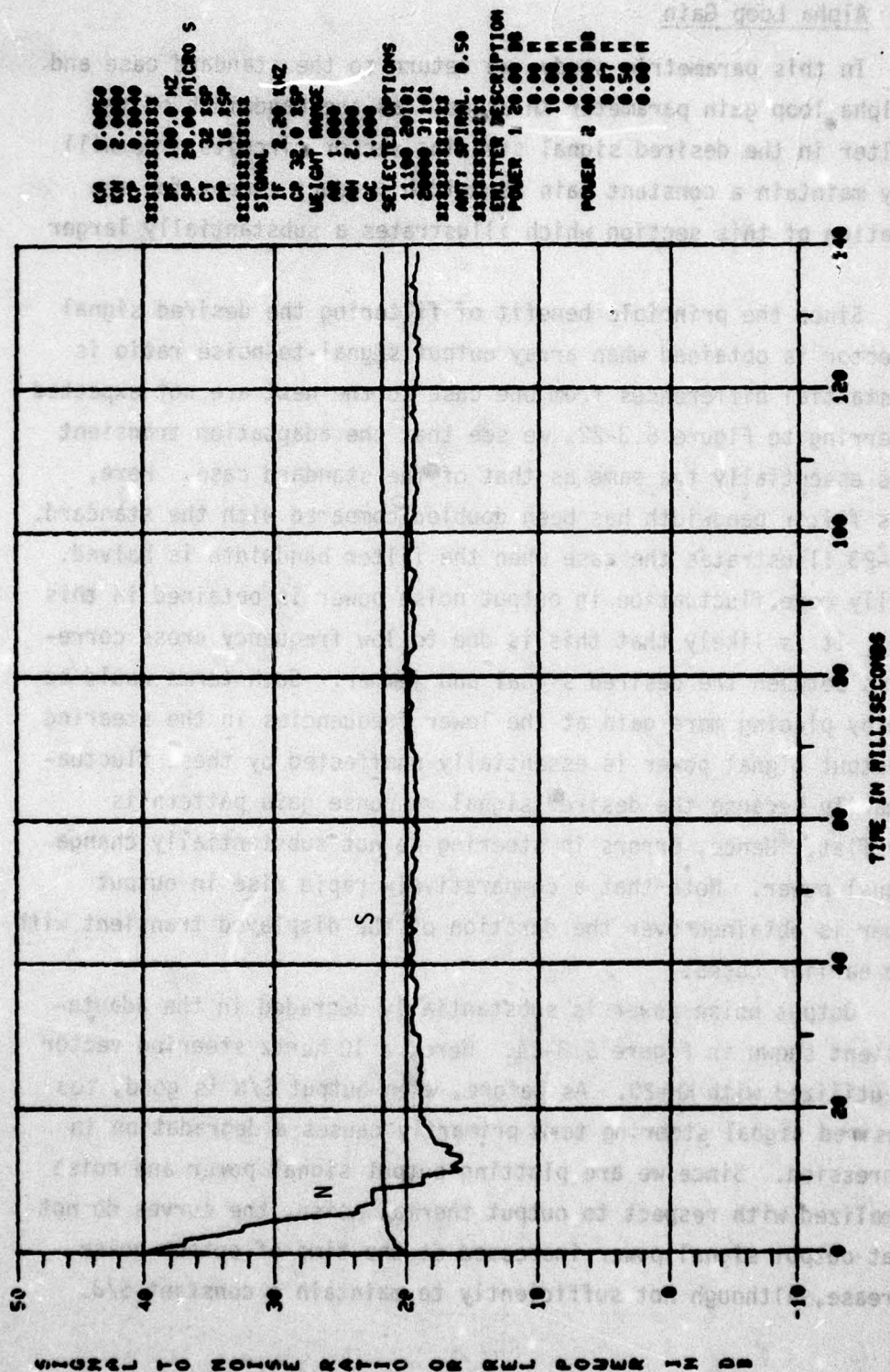


Figure 5.3-21. Adaptation Transients for a Perturbation Sequence Length of 16 and KP=0.004

5.3.4 Alpha Loop Gain

In this parametric study, we return to the standard case and vary the alpha loop gain parameter KH as well as the bandwidth of the lowpass filter in the desired signal steering vector circuits. We will arbitrarily maintain a constant gain bandwidth product except for the last simulation of this section which illustrates a substantially larger product.

Since the principle benefit of filtering the desired signal steering vector is obtained when array output signal-to-noise ratio is small, substantial differences from one case to the next are not expected here. Referring to Figure 5.3-22, we see that the adaptation transient obtained is essentially the same as that of the standard case. Here, the lowpass filter bandwidth has been doubled compared with the standard. Figure 5.3-23 illustrates the case when the filter bandwidth is halved. Substantially more fluctuation in output noise power is obtained in this later case. It is likely that this is due to low frequency cross correlation terms between the desired signal and jammer. Such terms would be emphasized by placing more gain at the lower frequencies in the steering vector. Output signal power is essentially unaffected by these fluctuations primarily because the desired signal response gain pattern is relatively flat. Hence, errors in steering do not substantially change output signal power. Note that a comparatively rapid rise in output signal power is obtained over the duration of the displayed transient with respect to earlier cases.

Output noise power is substantially degraded in the adaptation transient shown in Figure 5.3-24. Here, a 10 hertz steering vector filter is utilized with $KH=20$. As before, when output S/N is good, too large a desired signal steering term primarily causes a degradation in noise suppression. Since we are plotting output signal power and noise power normalized with respect to output thermal noise, the curves do not reveal that output signal power increased at the time of output noise power increase, although not sufficiently to maintain a constant S/J.

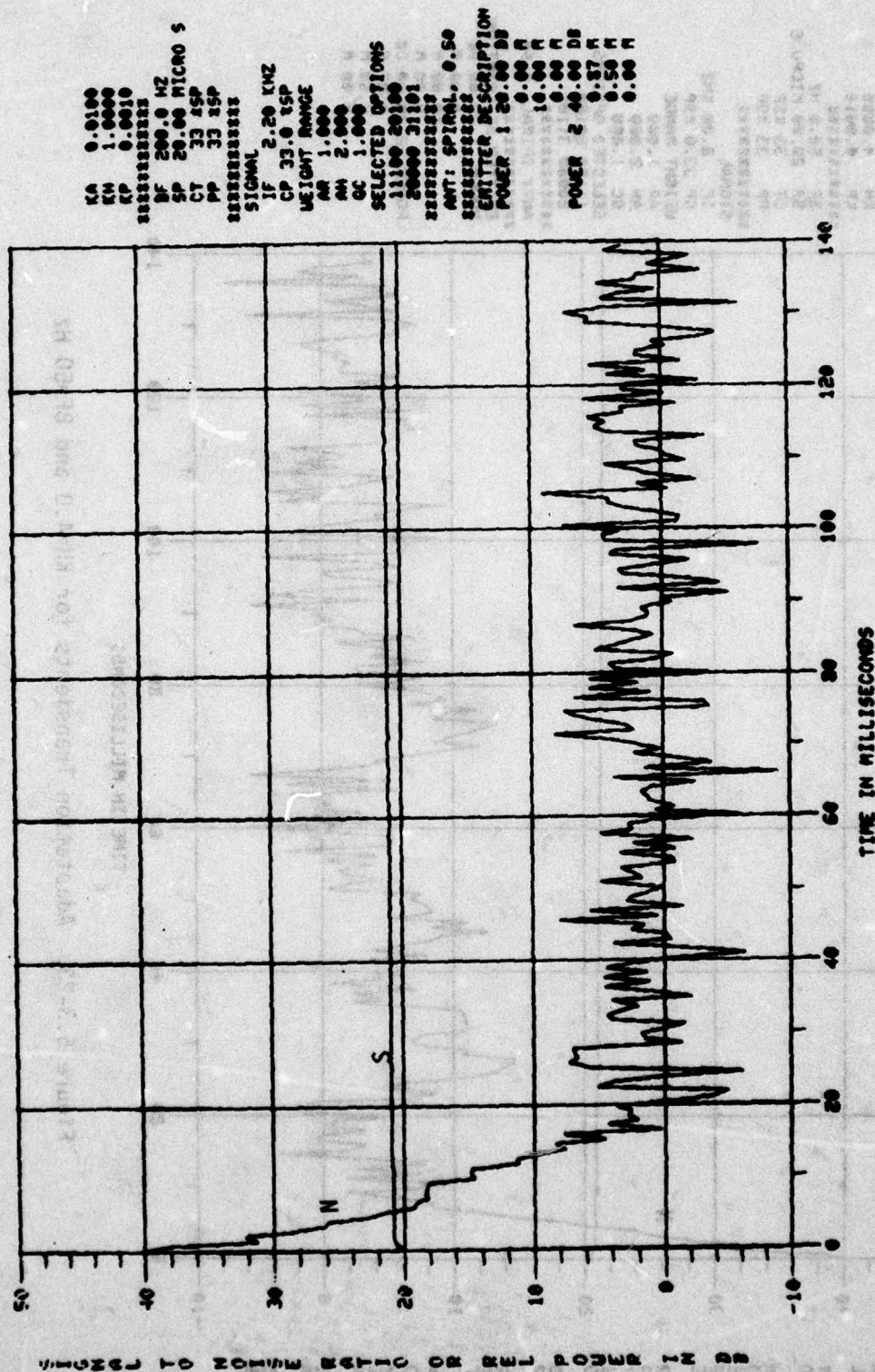


Figure 5.3-22. Adaptation Transients for KH=1.0 and BF=200 Hz

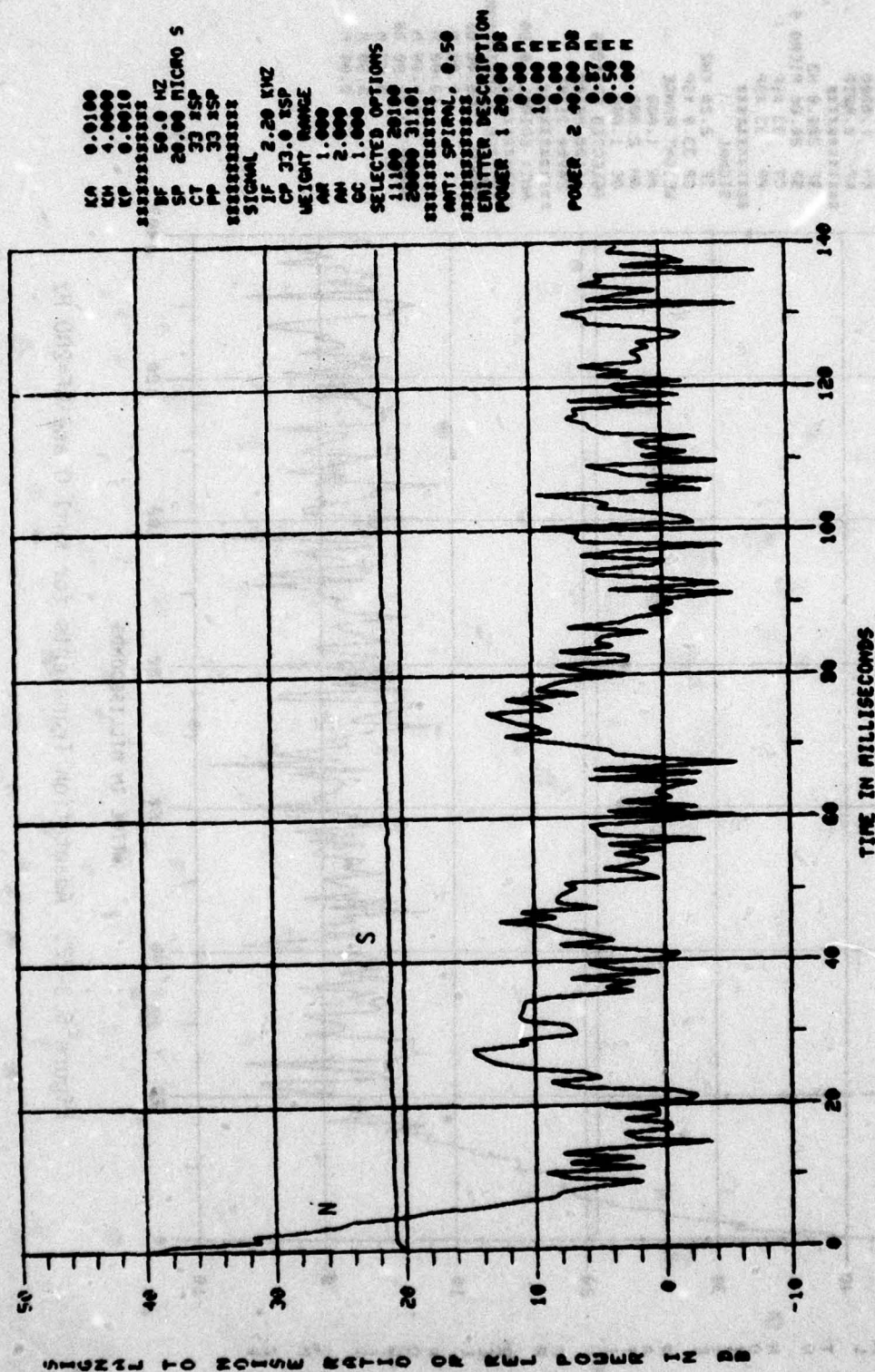


Figure 5.3-23. Adaptation Transients for KH=4.0 and BF=50 Hz

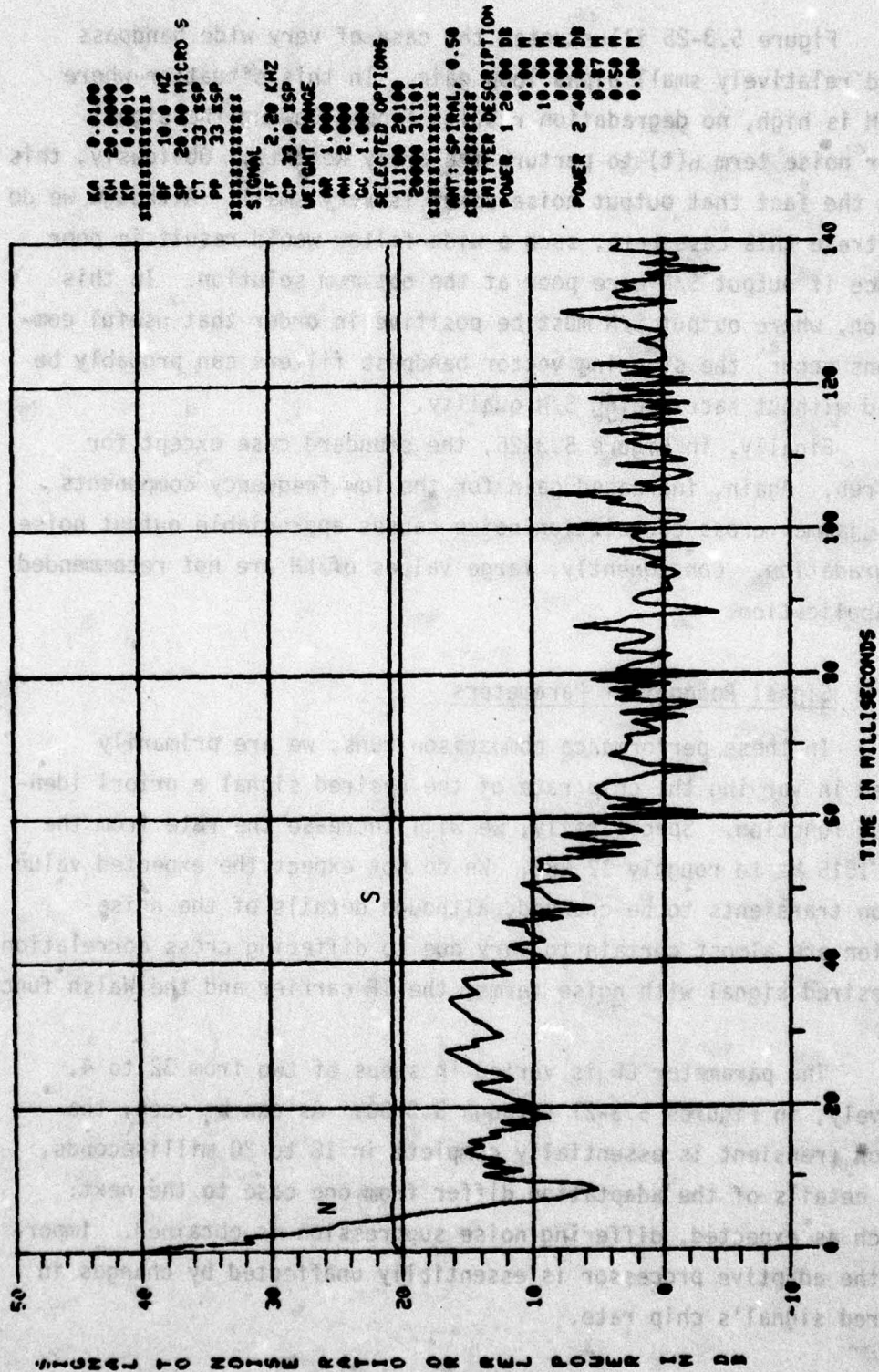


Figure 5.3-24. Adaptation Transients for KH=20 and BF=10 Hz

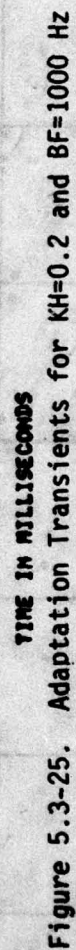
Figure 5.3-25 illustrates the case of very wide bandpass filter and relatively small alpha loop gain. In this situation where output S/N is high, no degradation results from allowing the signal recognizer noise term $n(t)$ to perturb the array weights. Obviously, this is due to the fact that output noise power is very small. Although we do not illustrate this case here, such a wide filter would result in poor performance if output S/N were poor at the optimum solution. In this application, where output S/N must be positive in order that useful communications occur, the steering vector bandpass filters can probably be eliminated without sacrificing S/N quality.

Finally, in Figure 5.3-26, the standard case except for $KH=10$ is run. Again, increased gain for the low frequency components of signal jammer cross correlation noise causes appreciable output noise power degradation. Consequently, large values of KH are not recommended in this application.

5.3.5 Signal Recognizer Parameters

In these performance comparison runs, we are primarily interested in varying the chip rate of the desired signal a priori identification function. Specifically, we will increase the rate from the standard 1515 Hz to roughly 12 kHz. We do not expect the expected value adaptation transients to be changed, although details of the noise suppression are almost certain to vary due to differing cross correlation of the desired signal with noise terms, the IF carrier and the Walsh functions.

The parameter CP is varied in steps of two from 32 to 4, respectively, in Figures 5.3-27 through 5.3-30. As can be seen, the adaptation transient is essentially complete in 18 to 20 milliseconds, although details of the adaptation differ from one case to the next. Also, much as expected, differing noise suppression is obtained. Importantly, the adaptive processor is essentially unaffected by changes in the desired signal's chip rate.



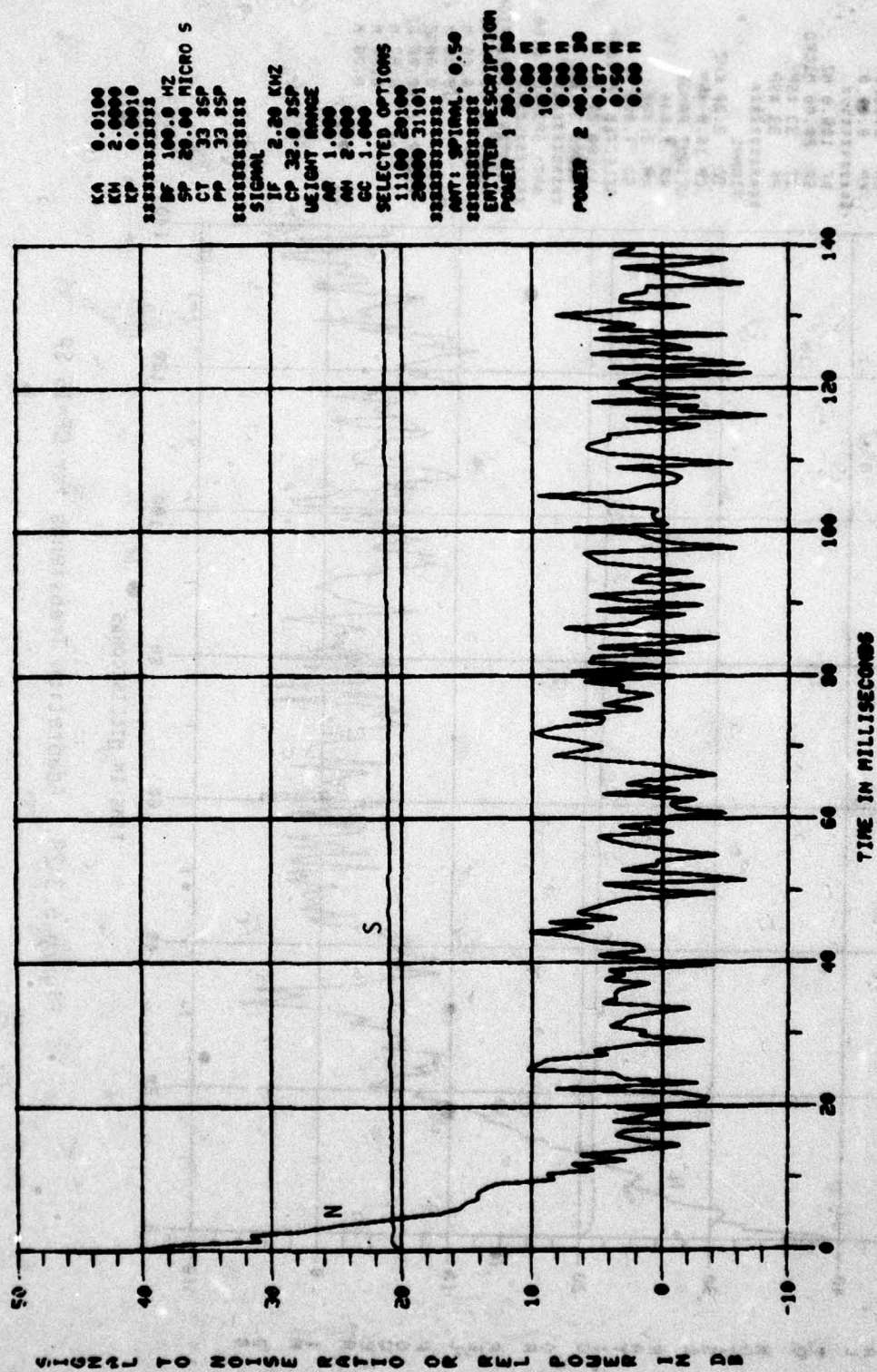


Figure 5.3-27. Adaptation Transients for CP=32 SP

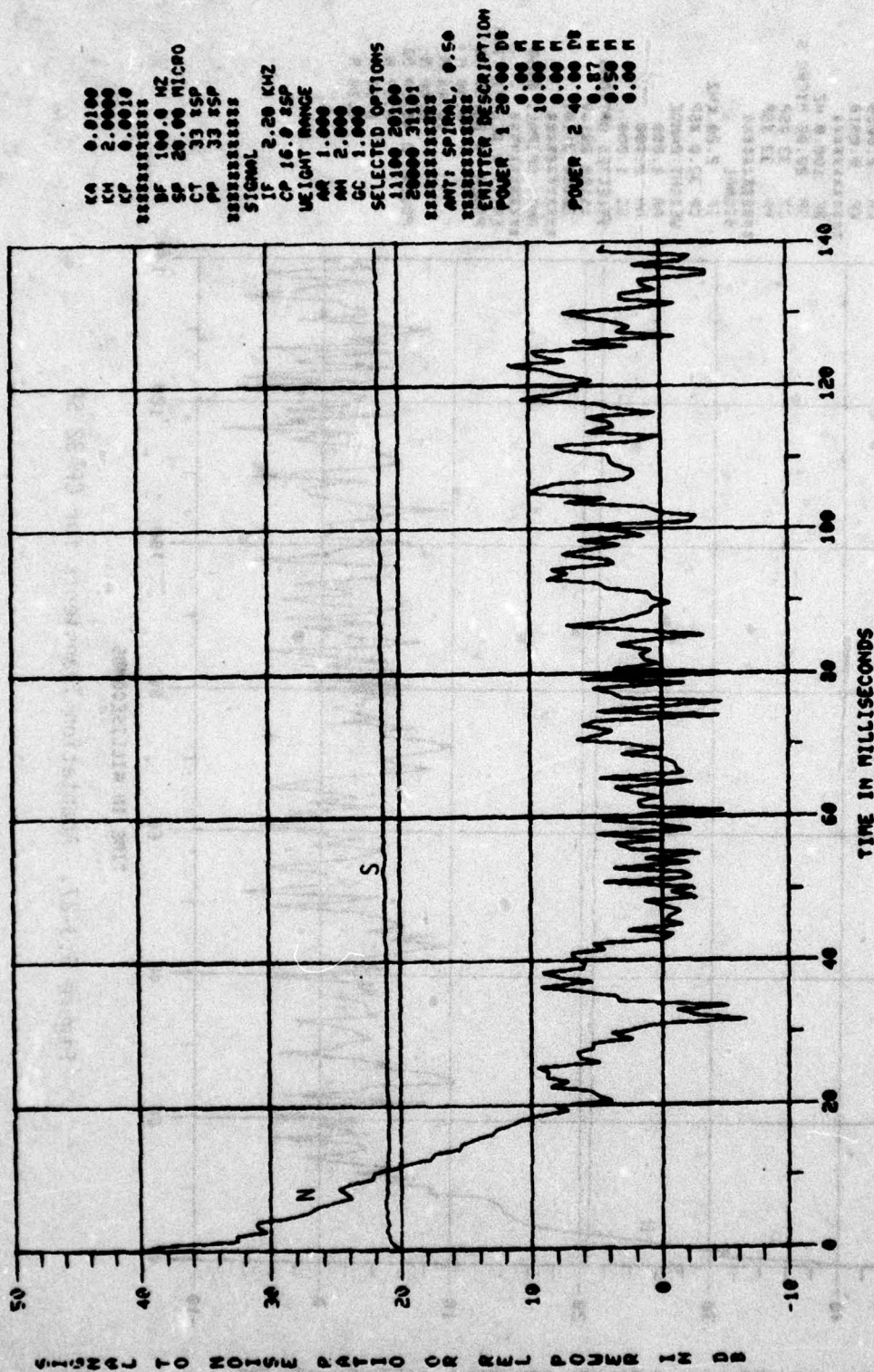
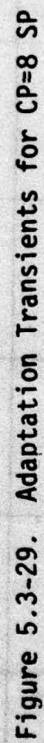


Figure 5.3-28. Adaptation Transients for CP=16 SP



The graph displays two waveforms, N and S, plotted against time in milliseconds. The vertical axis ranges from -10 to 50, and the horizontal axis ranges from 0 to 140. Waveform N is a smooth curve starting at 0, rising to a peak of about 35 at 20 ms, and then gradually declining. Waveform S is a highly oscillatory signal starting at 0, rising to a peak of about 35 at 20 ms, and then continuing with high-frequency oscillations between approximately 10 and 40 until 140 ms.

WIRE TO ZONEW BATH OR NEW BOWER IN AM

Next, we show the effect of signal recognizer delay not being an integral number of Walsh function cycles. According to the derivation in Appendix A, an incorrect desired signal gradient will result for this case. As seen in Figure 5.3-31, the adaptive processor fails to suppress the jammer, mistaking it as desired, and also substantially suppresses the desired signal.

Finally, let us examine an interesting property of pseudo random sequences. The term $C(t)C(t-\tau)$, where $C(t)$ is the a priori desired signal pseudo random discriminant, is important in that its integral establishes decorrelation of noise terms emerging for the signal recognizer. Short-term fluctuation of this term throughout the period of the sequence establishes the degree of short-term signal-jammer decorrelation. In order to determine whether or not there might be some best value of signal recognizer delay time τ , we have summed the above product of discriminant function times delayed discriminant function over a period of the sequence, in this case 1023 chips for a 10 bit shift register. It is well known, of course, that this sum will be equal to one over the entire interval, but the running sum can deviate substantially from unity. We see this in Figures 5.3-32, 33, and 34. The delay value is respectively 4, 8 and 32 chips. Observe that we have plotted the sum versus time.

At first glance, the curves may appear to be substantially different. The first is more above zero than below, while the second is predominantly negative over most of the interval. The third is more negative than not, but not nearly so much as the second case. Closer observation, however, will show that except for a vertical and horizontal displacement, these three curves are identically the same. For example, the peak in 5.3-32 occurring at $N \approx 550$ appears on 5.3-33 at $N \approx 630$, and again on 5.3-34 at $N \approx 50$. The reason is for this equivalence is readily understood.

Given a linear sequence generator it can be shown that

$$C(t)C(t-\tau) = C(t-\delta) \quad (5-15)$$

where τ is a non zero integer and δ is an integer determined by τ and the sequence length. Thus the three figures simply display the summation of $C(t)$ with different delays and initial conditions. In conclusion, there is no optimum delay value for τ .

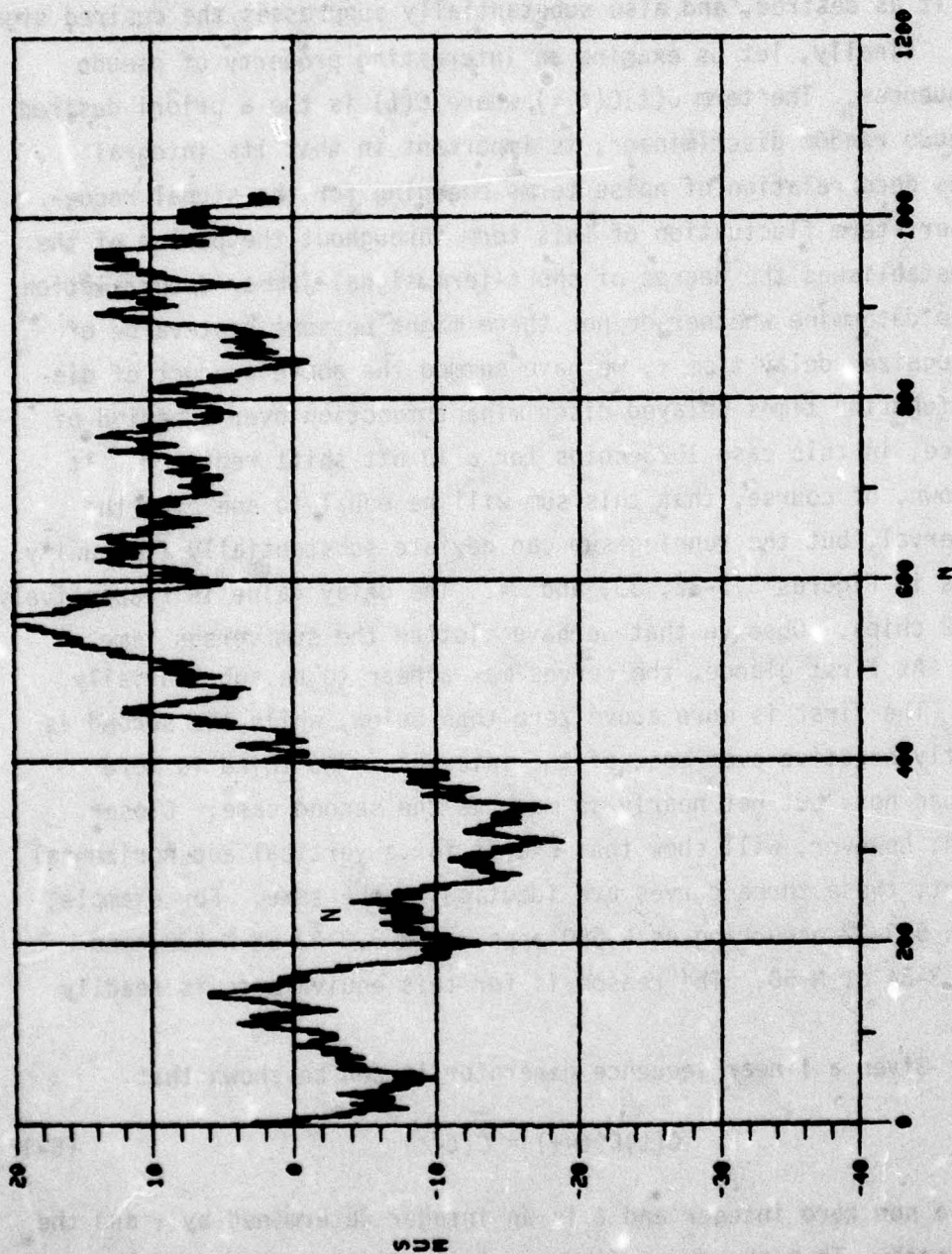


Figure 5.3-31. The Function $x_1 C(t) C(t-\tau)$ for $\tau=4$

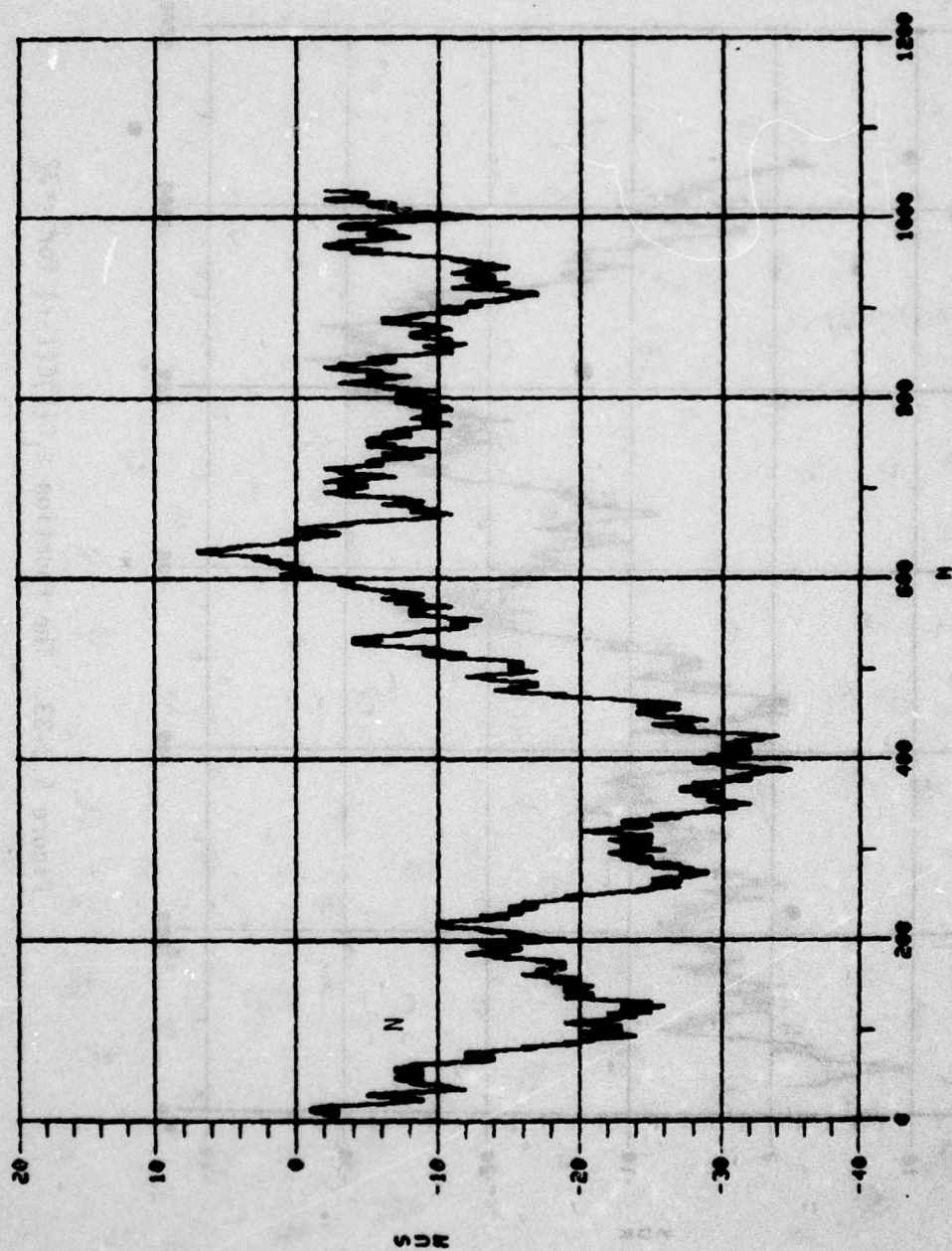


Figure 5.3-32. The Function $\sum_1 C(t)C(t-\tau)$ for $\tau=8$

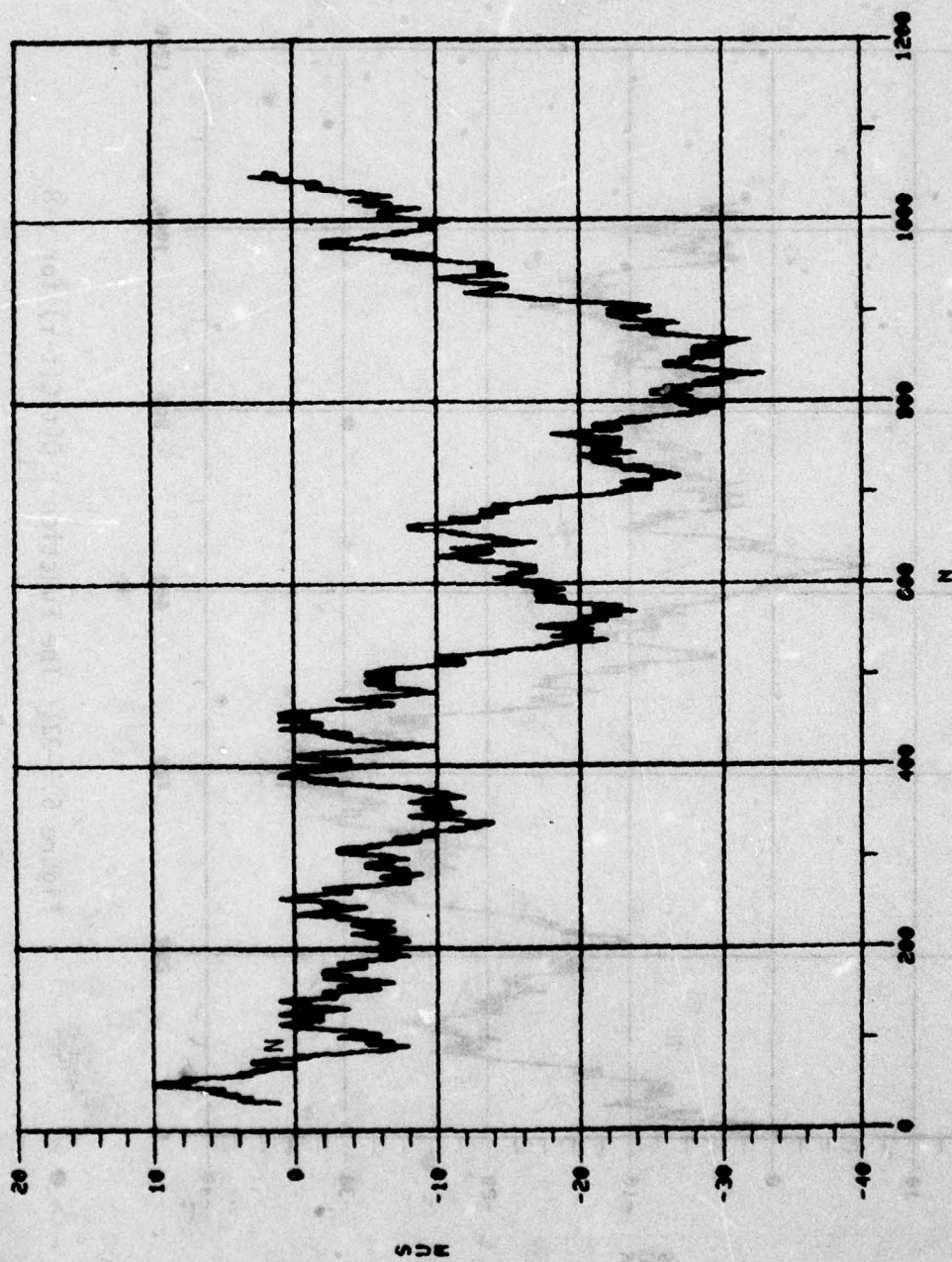


Figure 5.3-33. The Function $S_1 C(t) C(t-\tau)$ for $\tau=32$

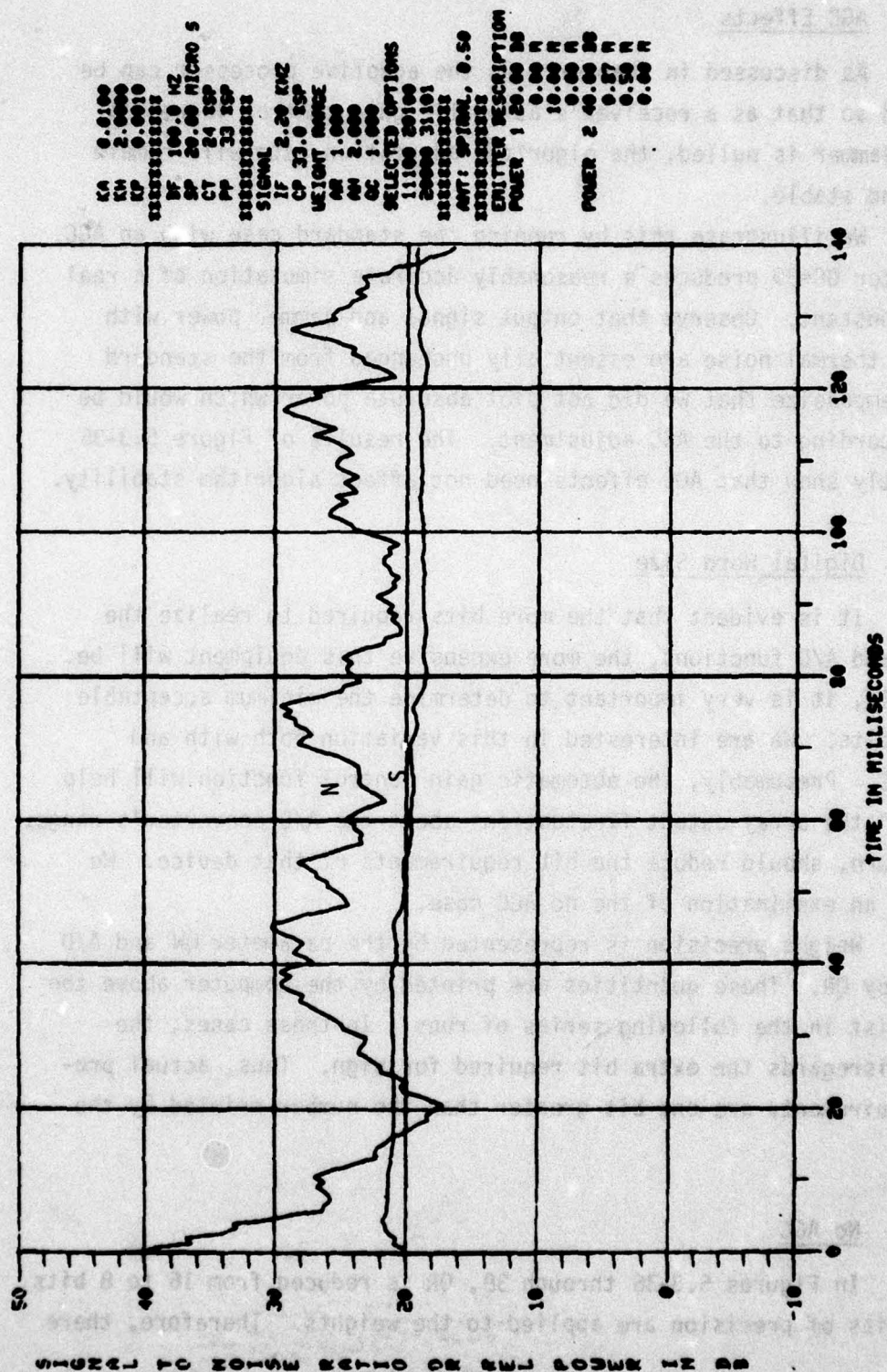


Figure 5.3-34. Adaptation Transients for Signal Recognizer Delay Incorrectly Chosen

5.3.6 AGC Effects

As discussed in Chapter 4.0, the adaptive processor can be compensated so that as a receiver's automatic gain control increases gain as a jammer is nulled, the algorithm adaptation rate will remain constant and stable.

We illustrate this by running the standard case with an AGC. The parameter GC=39 produces a reasonably accurate simulation of a real AGC time constant. Observe that output signal and jammer power with respect to thermal noise are essentially unchanged from the standard case. We emphasize that we did not plot absolute power which would be varying according to the AGC adjustment. The results of Figure 5.3-35 significantly show that AGC effects need not affect algorithm stability.

5.3.7 Digital Word Size

It is evident that the more bits required to realize the weighting and A/D functions, the more expensive this equipment will be. Consequently, it is very important to determine the minimum acceptable number of bits. We are interested in this variation both with and without AGC. Presumably, the automatic gain control function will help to "center" the array output fluctuations about the A/D converter's range. This, in turn, should reduce the bit requirements of that device. We begin with an examination of the no AGC case.

Weight precision is represented by the parameter QW and A/D precision by QR. These quantities are printed by the computer above the standard list in the following series of runs. In these cases, the computer disregards the extra bit required for sign. Thus, actual precision requirements are one bit greater than the number printed by the computer.

5.3.7.1 No AGC

In Figures 5.3-36 through 38, QR is reduced from 16 to 8 bits while 24 bits of precision are applied to the weights. Therefore, there

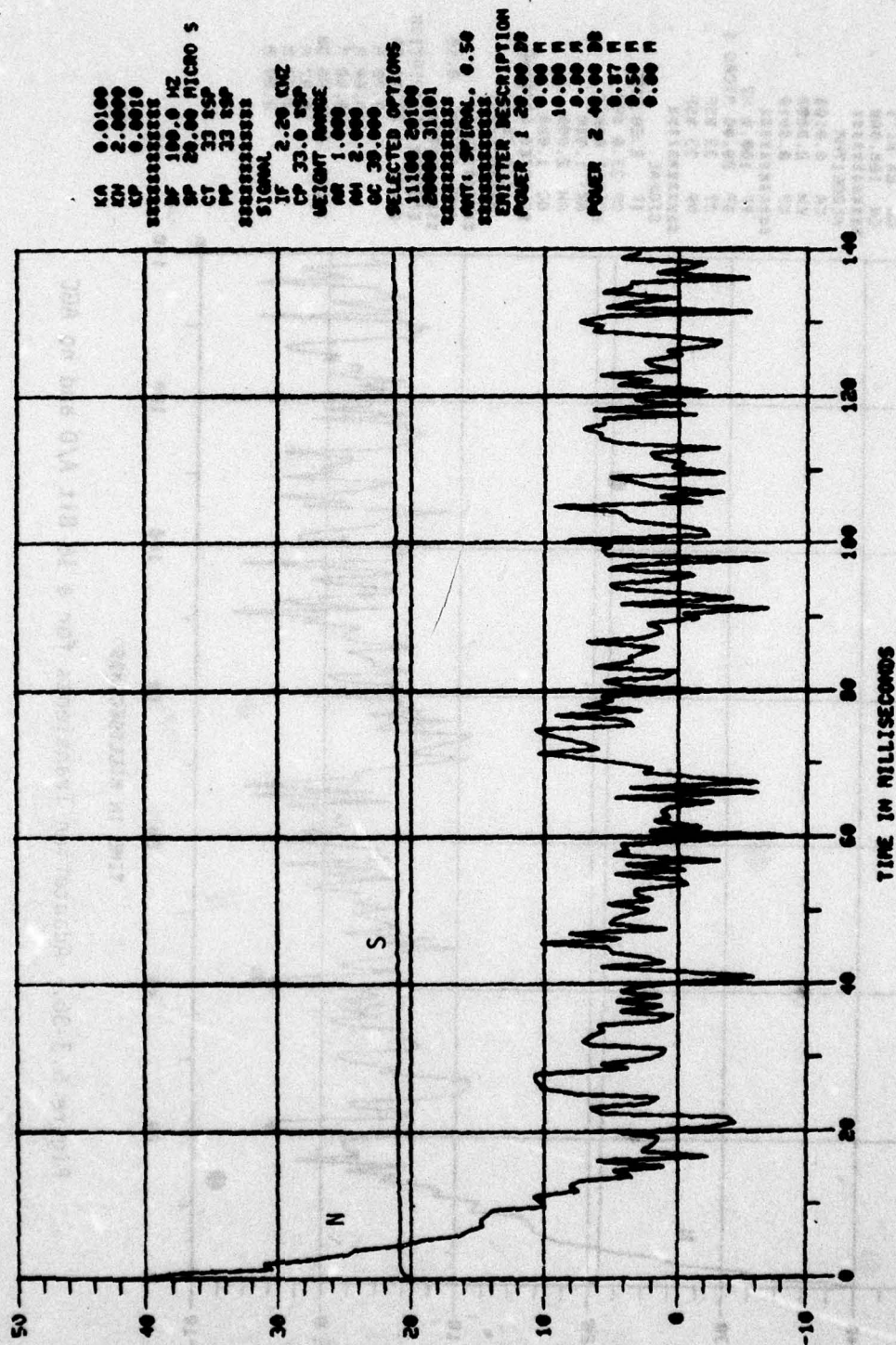


Figure 5.3-35. Adaptation Transients for the Standard Case Including AGC Compensation

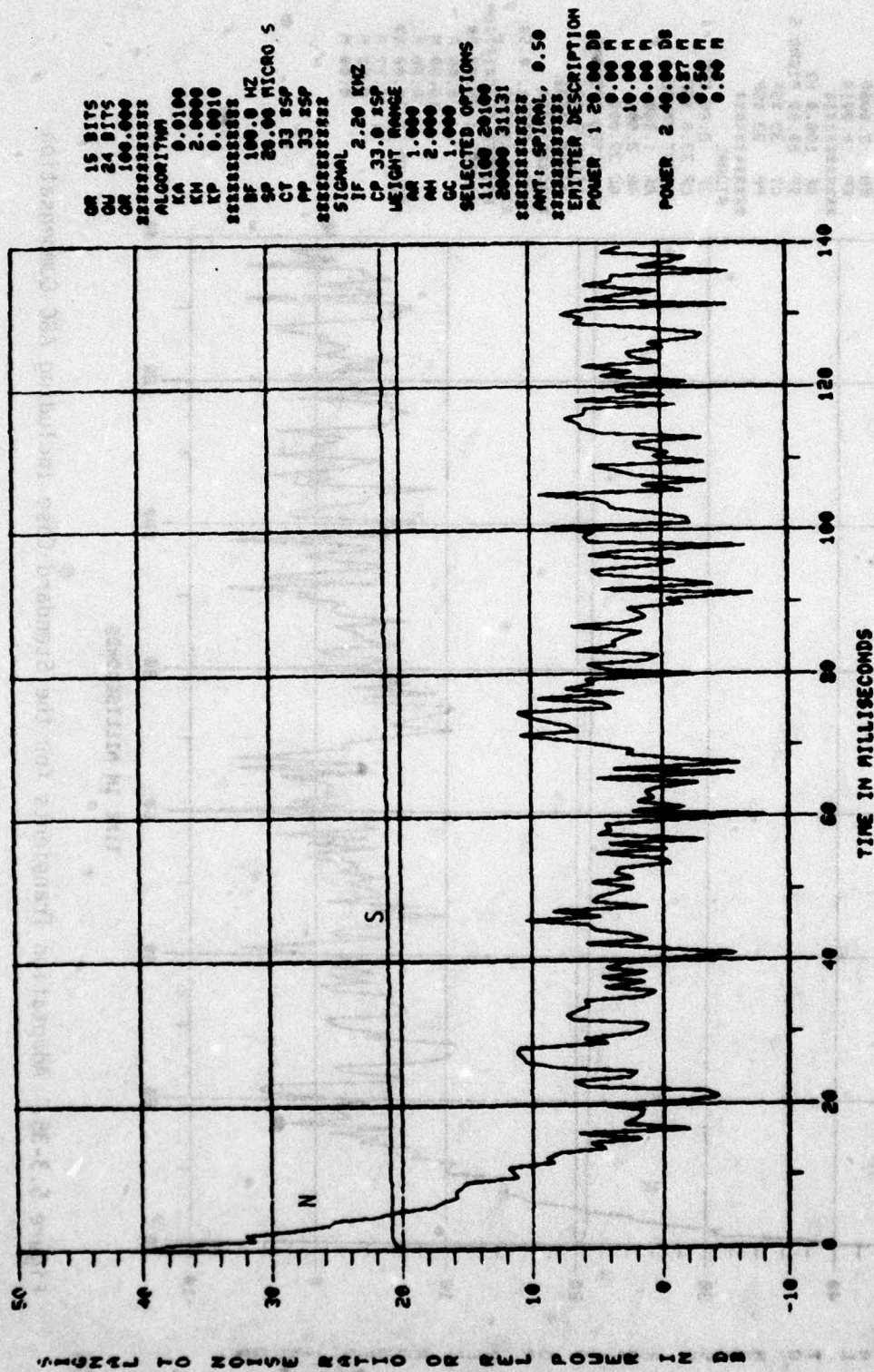


Figure 5.3-36. Adaptation Transients for a 16 Bit A/D and no AGC

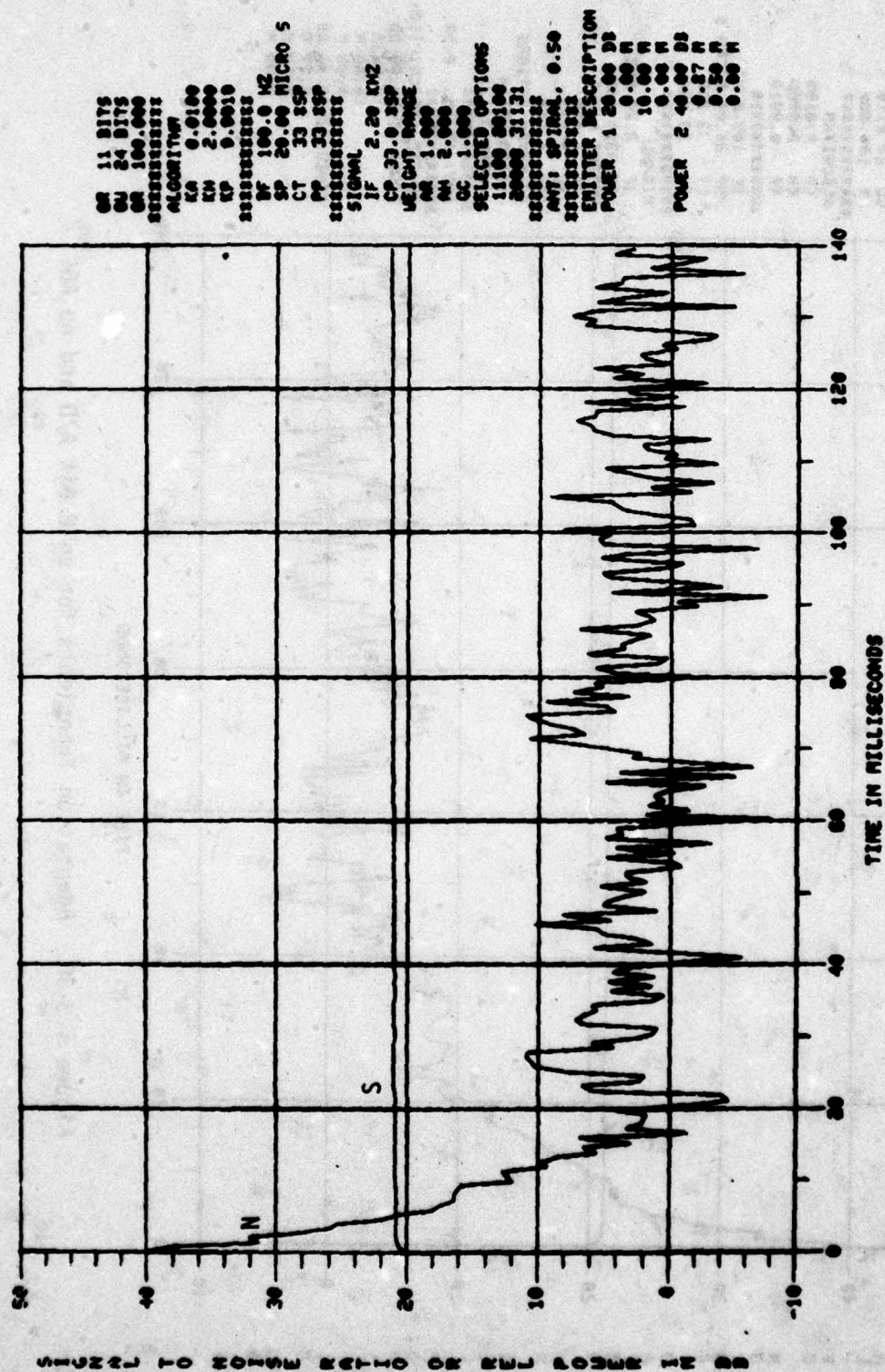


Figure 5.3-37. Adaptation Transients for a 12 Bit A/D and no AGC

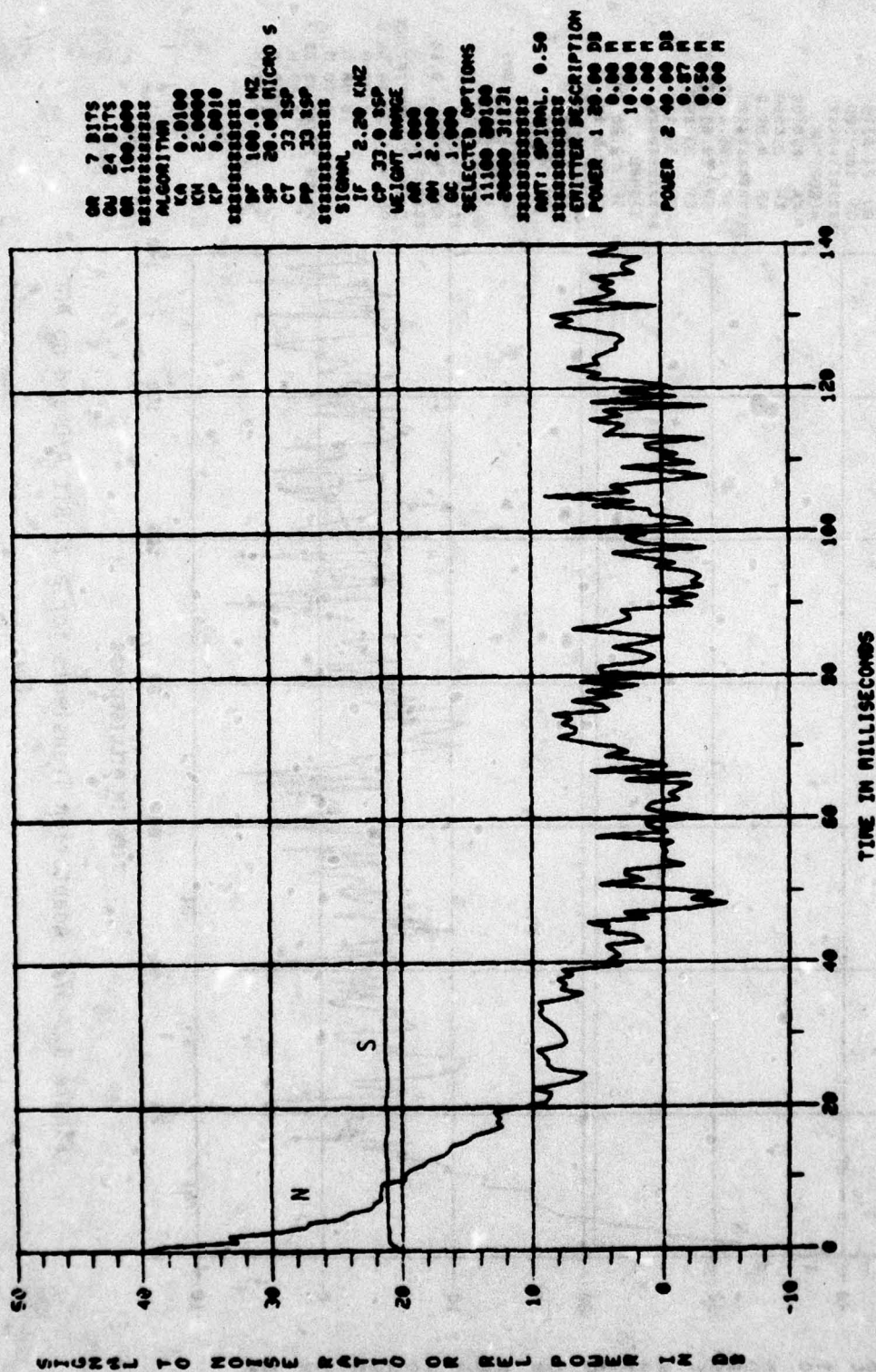


Figure 5.3-38. Adaptation Transients for an 8 Bit A/D and no AGC

is essentially no degradation in adaptive array performance due to weight precision. In these runs of the standard case, it is evident that QR as small as 12 bits results in no perceivable degradation in the adaptation transient. An acceptable transient is obtained even in the case of QR equal to 8 bits. Reduced adaptation speed in this later case is probably due to errors in gradient estimation.

In the next set of runs, Figures 5.3-39 through 41, the A/D converter precision is held constant at 24 bits and the weight precision is reduced from 16 bits to 8 bits (listed on the output as 15 bits, 11, and 7 bits). Again, the 12-bit case shows essentially no change from the standard case. Substantial difference is seen for QW=8 bits although adaptation is occurring. In a future investigation, it would be desirable to run this later case for long values of time so as to determine whether or not approximately optimum solutions were obtained.

Next, in Figure 5.3-42, we examine the important case of QR and QW equal to 12 bits each. This is a practical value for both weight and A/D converter precision. As can be seen, the adaptation transient is essentially identical to that of the standard case.

Figure 5.3-43 illustrates the case of A/D converter precision equal to 12 bits and weight precision equal to 8 bits. This case may be compared to the one given in Figure 5.3-41 for essentially infinite A/D converter precision. It is evident that the poor adaptation results seen in 5.3-43 are due to a compounding of A/D and weight precision errors since this case is considerably poorer than that shown in 5.3-41.

Additional investigation needs to be done regarding weight precision effects in order to obtain a good engineering tradeoff between required A/D weight precision and adaptive array performance.

5.3.7.2 With AGC

Presumably, the A/D converter precision requirement can be relaxed if the AGC operates sufficiently fast to essentially center receiver output fluctuations within the A/D converter range. Therefore, an investigation of bit precision was made for a nominal AGC. In

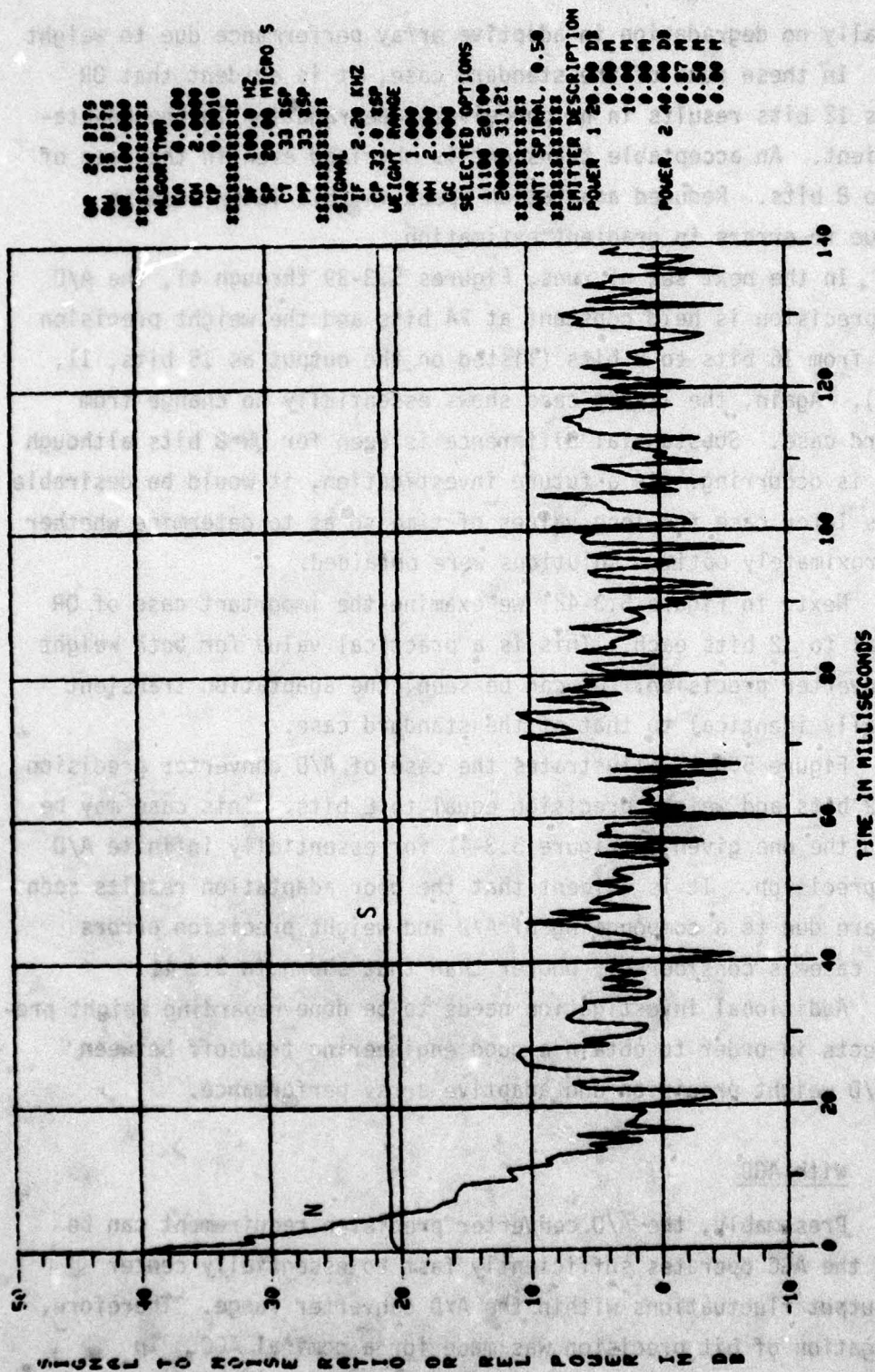


Figure 5.3-39. Adaptation Transients for a 16 Bit Weight and no AGC

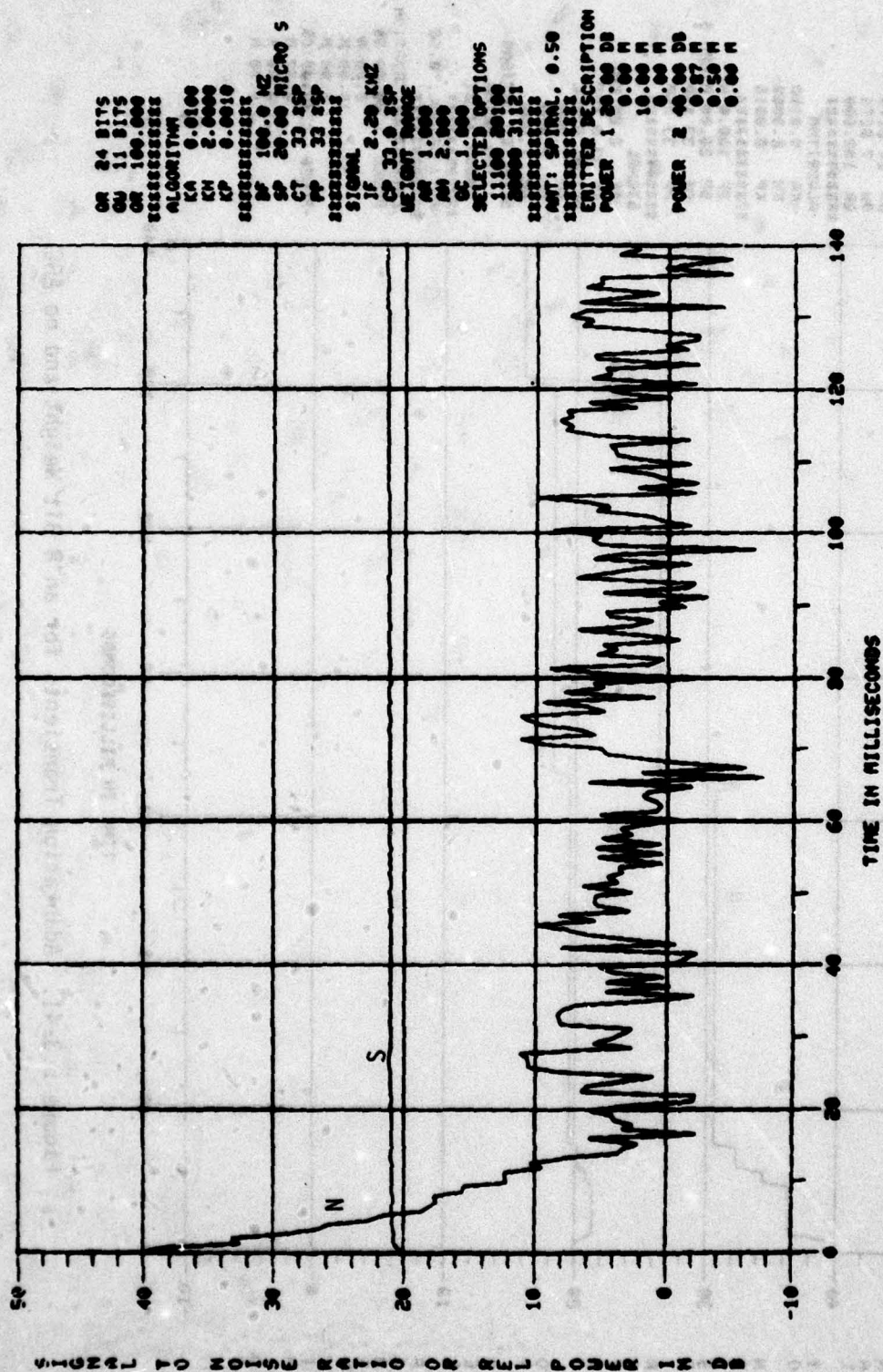
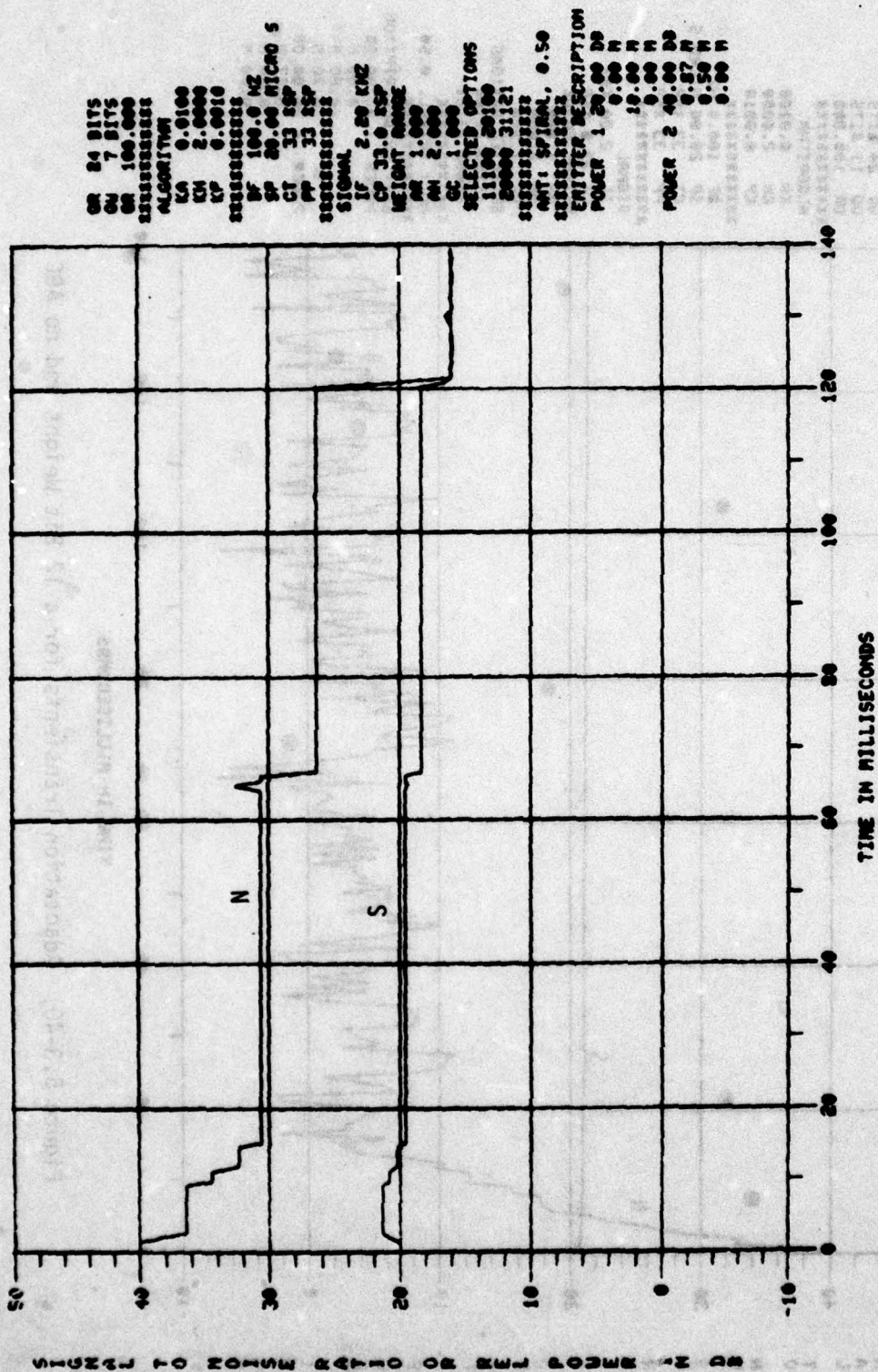
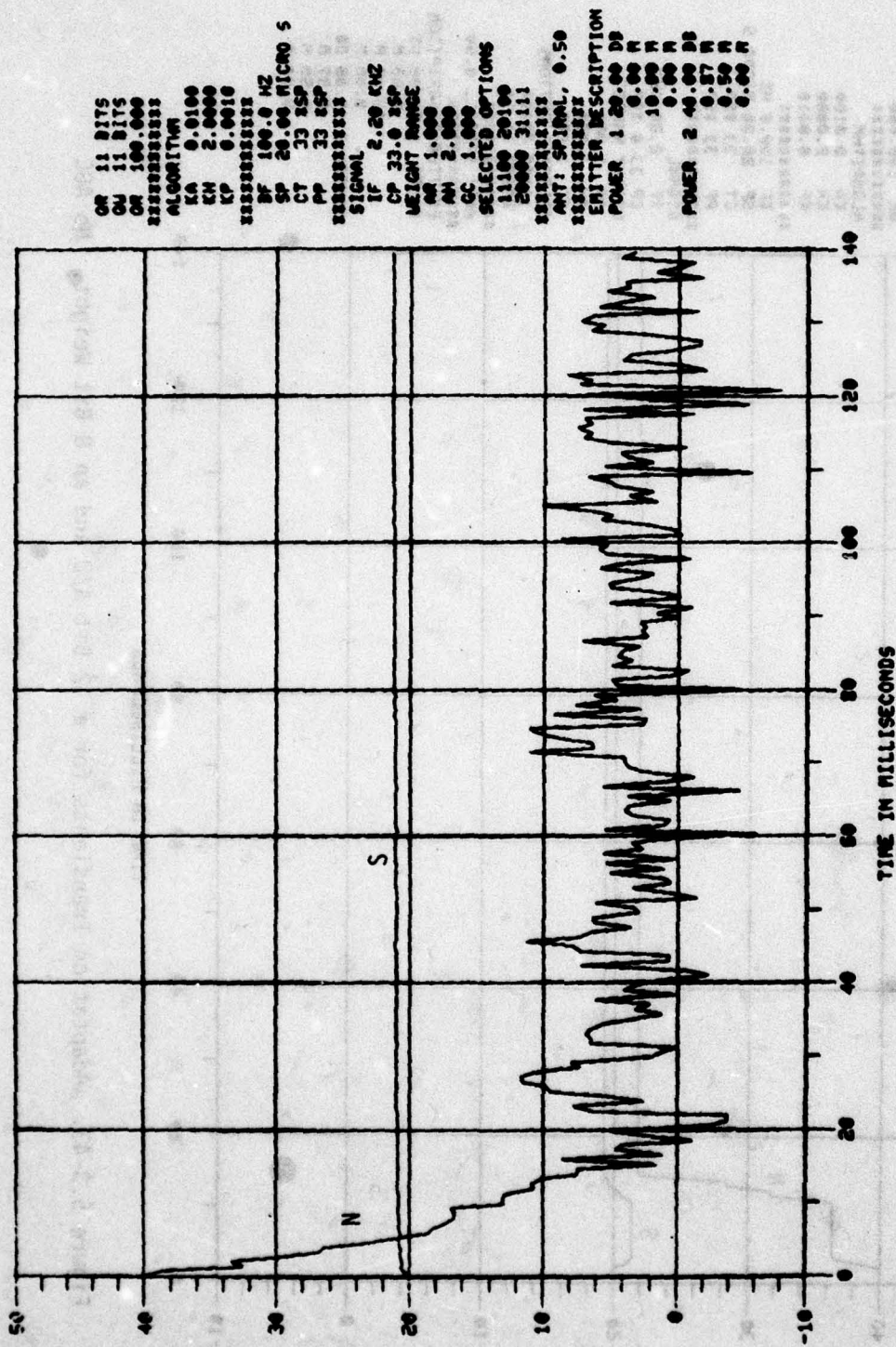


Figure 5.3-40. Adaptation Transients for a 12 Bit Weight and no AGC



OR 84 BITS
 ON 7 BITS
 ON 100.000
 =====
 ALGORITHM
 LA 0.0100
 LN 2.0000
 LP 0.0010
 =====
 SF 100.0 KHZ
 SP 20.00 MICRO S
 CT 33 KSP
 CP 33 KSP
 =====
 SIGNAL
 IF 2.00 KHZ
 CP 33.0 KSP
 HEIGHT RANGE
 AR 1.000
 AN 2.000
 GC 1.000
 =====
 SELECTED OPTIONS
 1100 20100
 20000 31121
 =====
 ANT. SPINAL, 0.50
 =====
 EMITTER DESCRIPTION
 POWER 1 20.00 DB
 0.00 N
 10.00 N
 0.00 N
 POWER 2 40.00 DB
 0.00 N
 0.87 N
 0.50 N
 0.00 N

Figure 5.3-41. Adaptation Transients for an 8 Bit Weight and no AGC



OR 11 BITS
 OR 11 BITS
 OR 100.000
 ALGORITHM
 KA 0.0100
 KW 2.0000
 KP 0.0010
 SF 100.0 HZ
 SP 20.00 MICRO S
 CT 33 SSP
 PP 33 SSP
 SIGNAL
 IF 2.20 KHZ
 CP 33.0 SSP
 WEIGHT RANGE
 OR 1.000
 AN 2.000
 GC 1.000
 SELECTED OPTIONS
 11100 20100
 20000 31111
 ANT: SPIRAL, 0.50
 ERITTER DESCRIPTION
 POWER 1 20.00 DB
 0.00 N
 10.00 N
 0.00 N
 POWER 2 40.00 DB
 0.57 N
 0.50 N
 0.00 N

Figure 5.3-42. Adaptation Transients for a 12 Bit A/D and a 12 Bit Weight. No AGC.

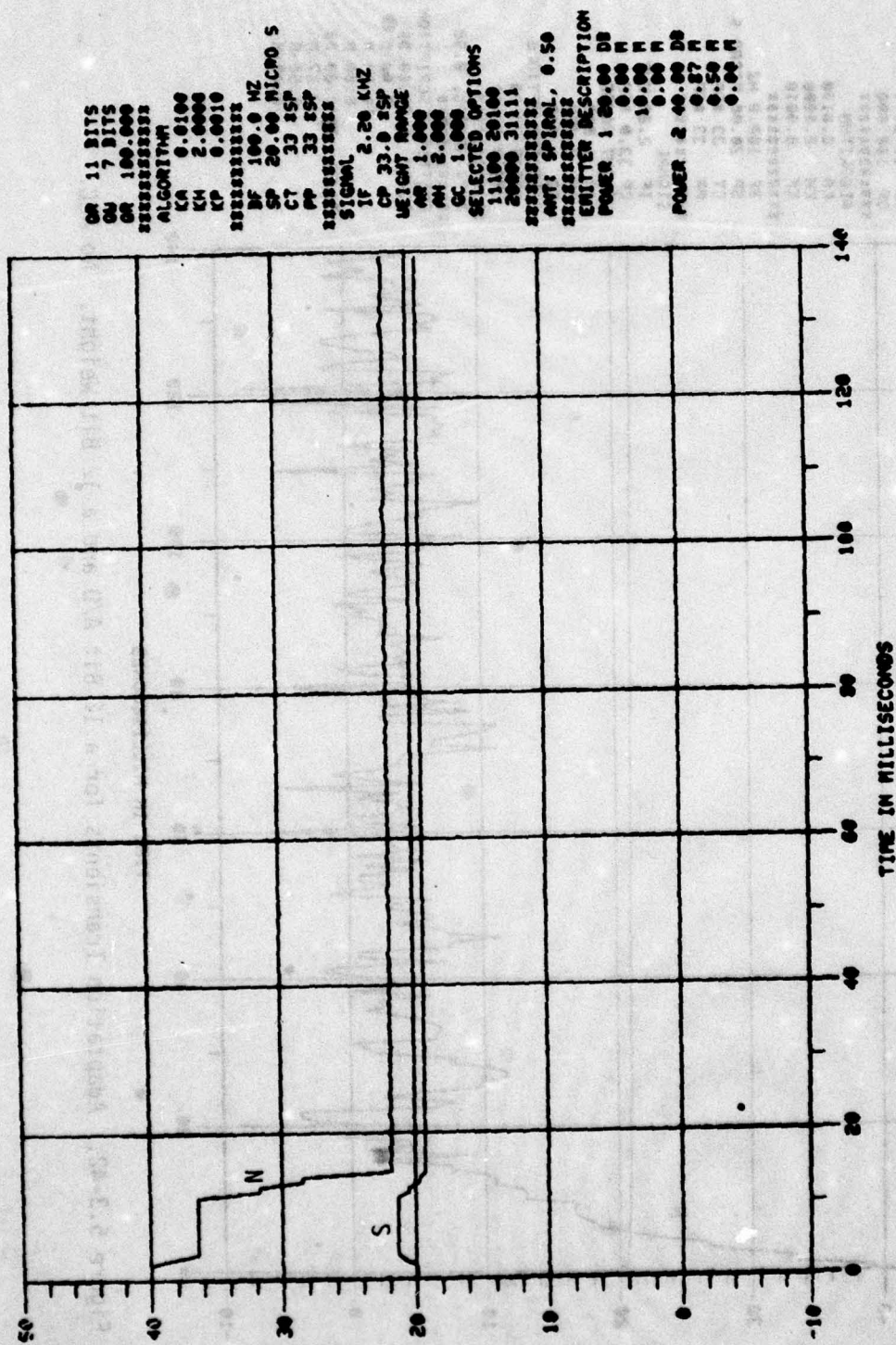


Figure 5.3-43. Adaptation Transients for a 12 Bit A/D and an 8 Bit Weight. No AGC

Figures 5.3-44 and 5.3-45, the A/D precision was held constant at 12 bits while weight precision was allowed to vary from 12 to 8 bits. As before, the 12/12 bit case gave predictable results. The 8 bit weight case was practically as good as when the A/D converter had infinite precision.

Figure 5.3-45 illustrates the case when weight precision is 8 bits and A/D converter precision is 8 bits. As can be seen, performance is very poor. The reason for this is probably that a typical receiver AGC operates much too slowly to provide any benefits in centering the receiver output for the A/D converter.

5.3.8 Rotation Effects

In a practical application, the adaptive array is almost always subjected to relative motion with respect to its environment. How well the adaptive processor can maintain a solution is dependent upon its gain parameters. In the following runs, the adaptive array is rotated with respect to the emitters at constant rates. In each case, rotation is about the Z axis (incorrectly identified on the graph as k_x). Figure 5.3-47 illustrates the results obtained when the rotation rate was 10^0 per second. Essentially no change is seen with respect to the standard case. Some degradation is seen in Figure 5.3-48 where the rotation rate is 100^0 per second. Observe that noise power has increased while signal power is decreasing, due to the motion of antenna pattern lobes past the desired signal. Adaptation rates for the desired signal are insufficient to maintain an essentially constant signal performance. Finally, a rotation rate of 240^0 per second is given in 5.3-49. Although the array continues to realize roughly 20 dB performance gain with respect to the isotropic element case, output signal-to-noise ratio is near zero or negative over a large portion of time. The adaptive processor is clearly too slow to maintain greater than 20 dB J/S improvement for this rotation rate.

By way of comparison, only very high performance aircraft will roll at rates greater than 100^0 per second. Even in this case,



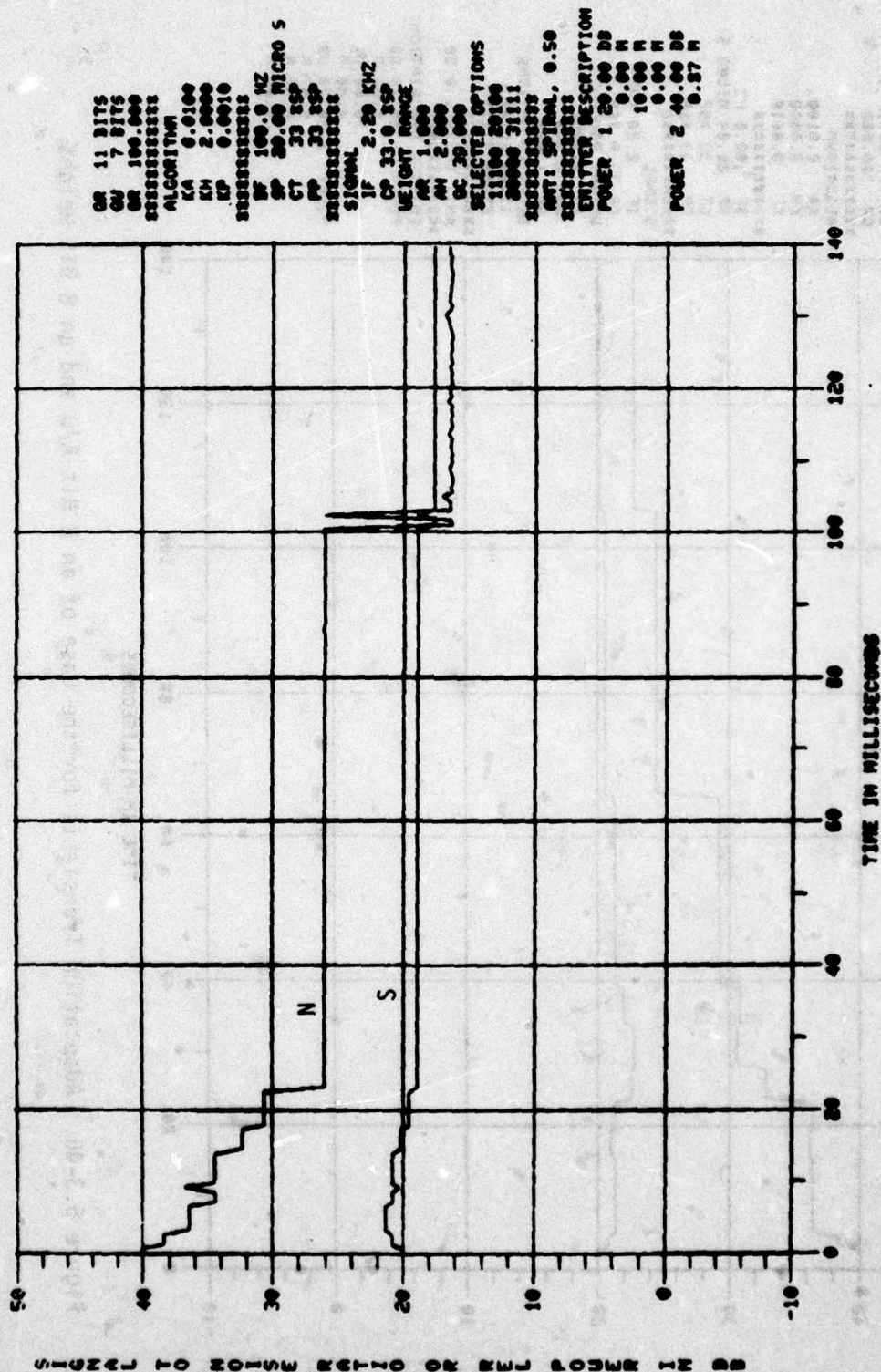


Figure 5.3-45. Adaptation Transients for a 12 Bit A/D and an 8 Bit Weight with AGC

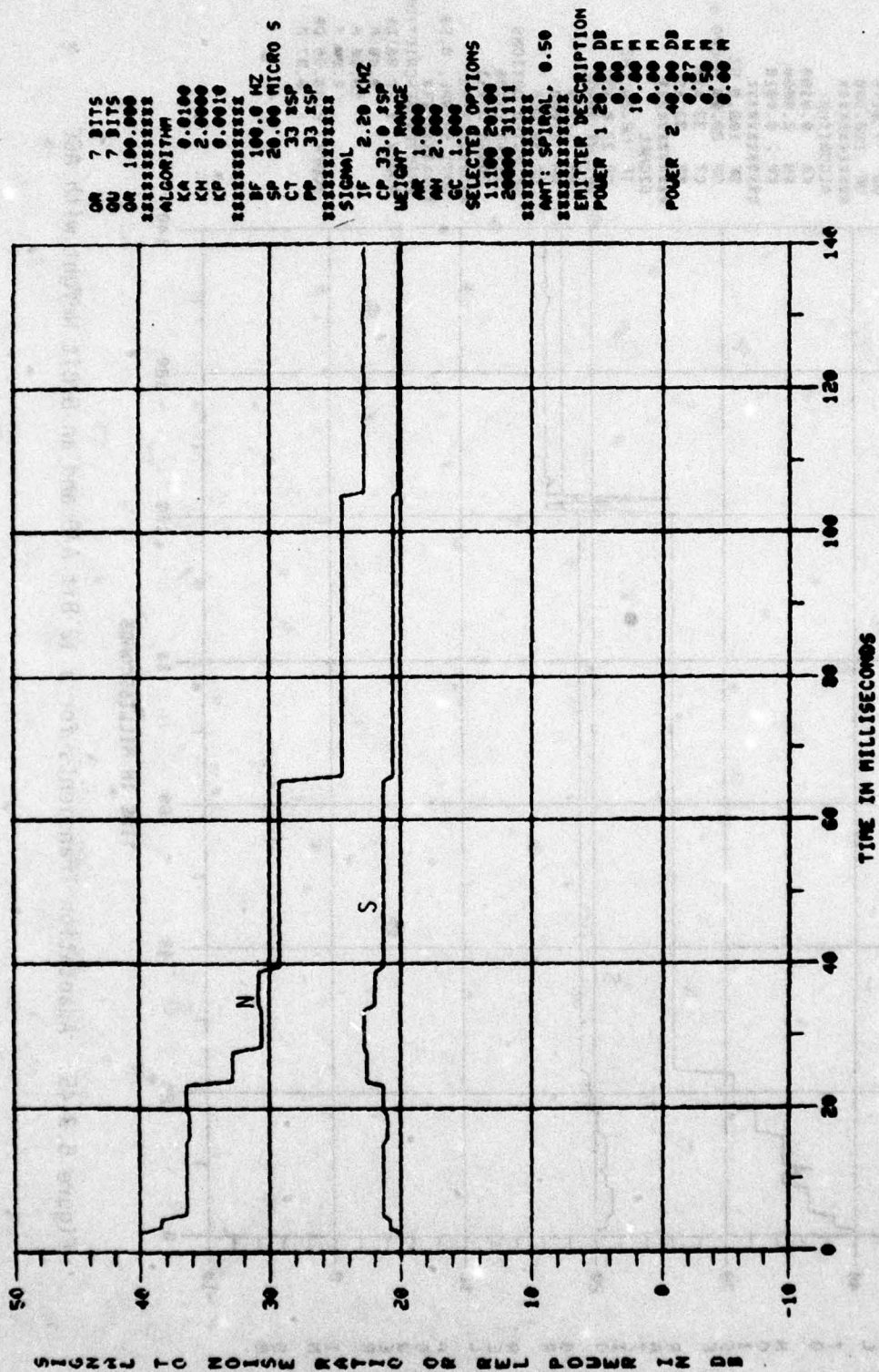


Figure 5.3-46. Adaptation Transients for the Case of an 8 Bit A/D and an 8 Bit Weight

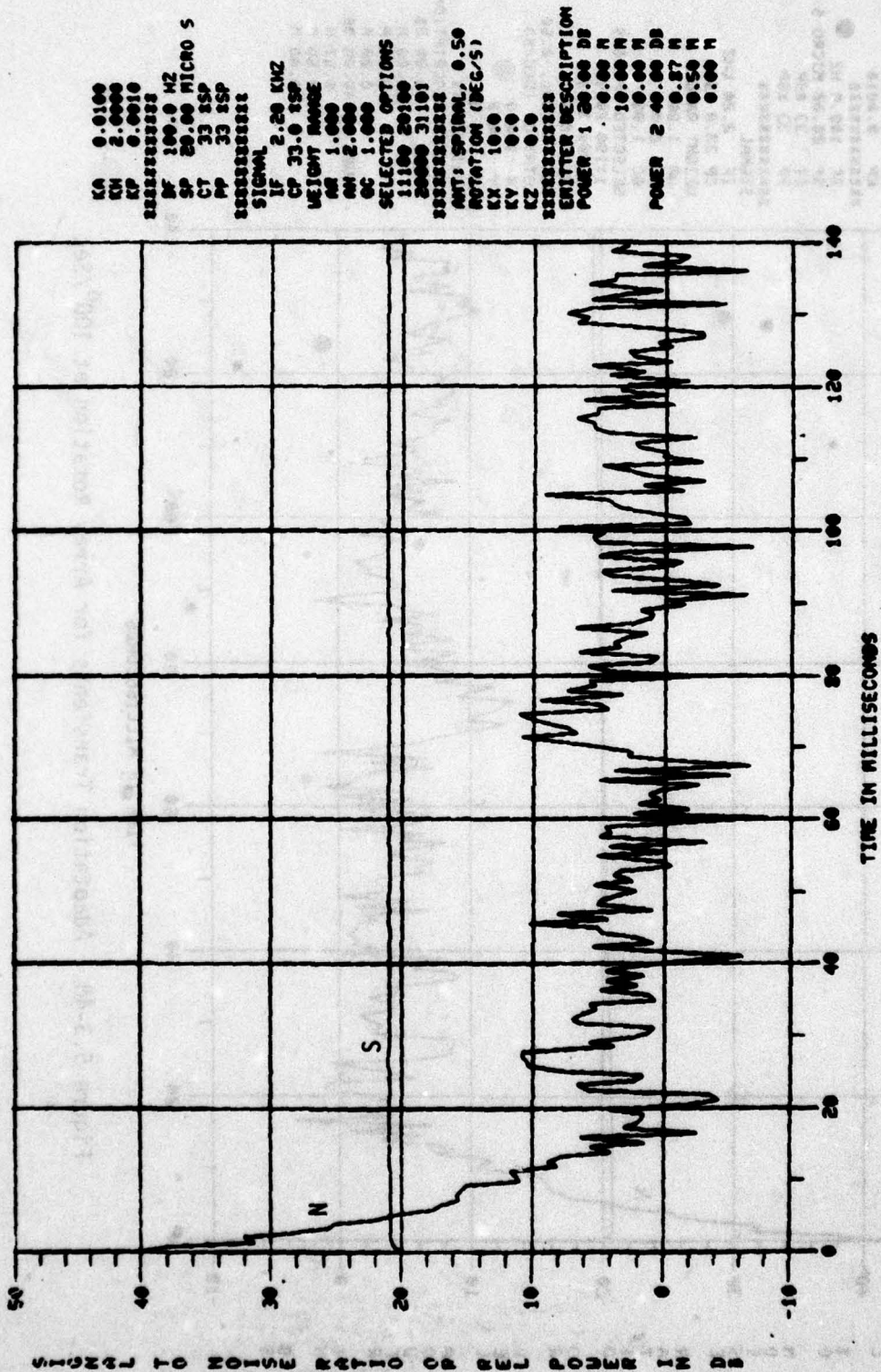


Figure 5.3-47. Adaptation Transients for Array Rotating at 10°/Sec

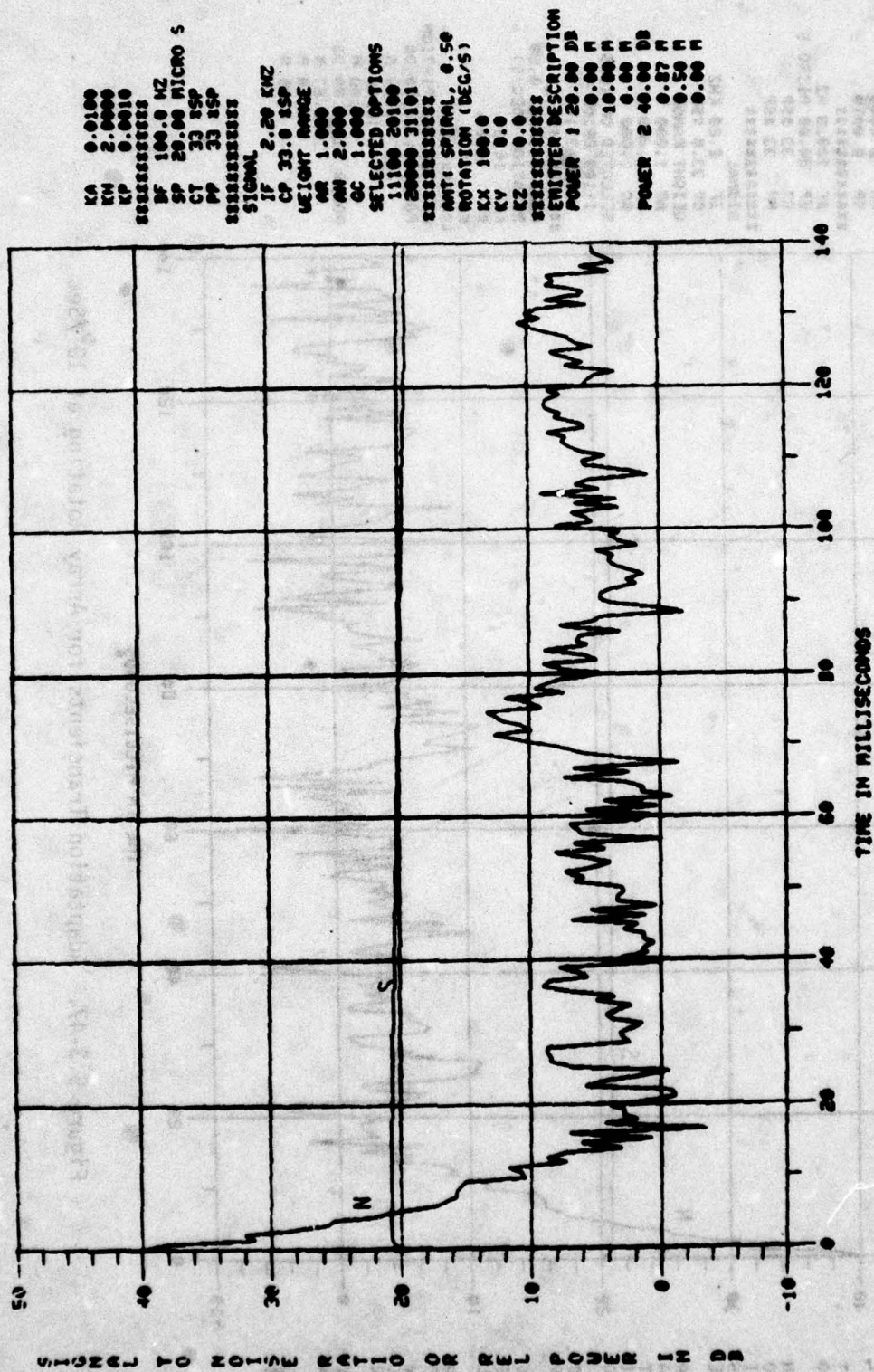


Figure 5.3-48. Adaptation Transients for Array Rotation at 100°/Sec

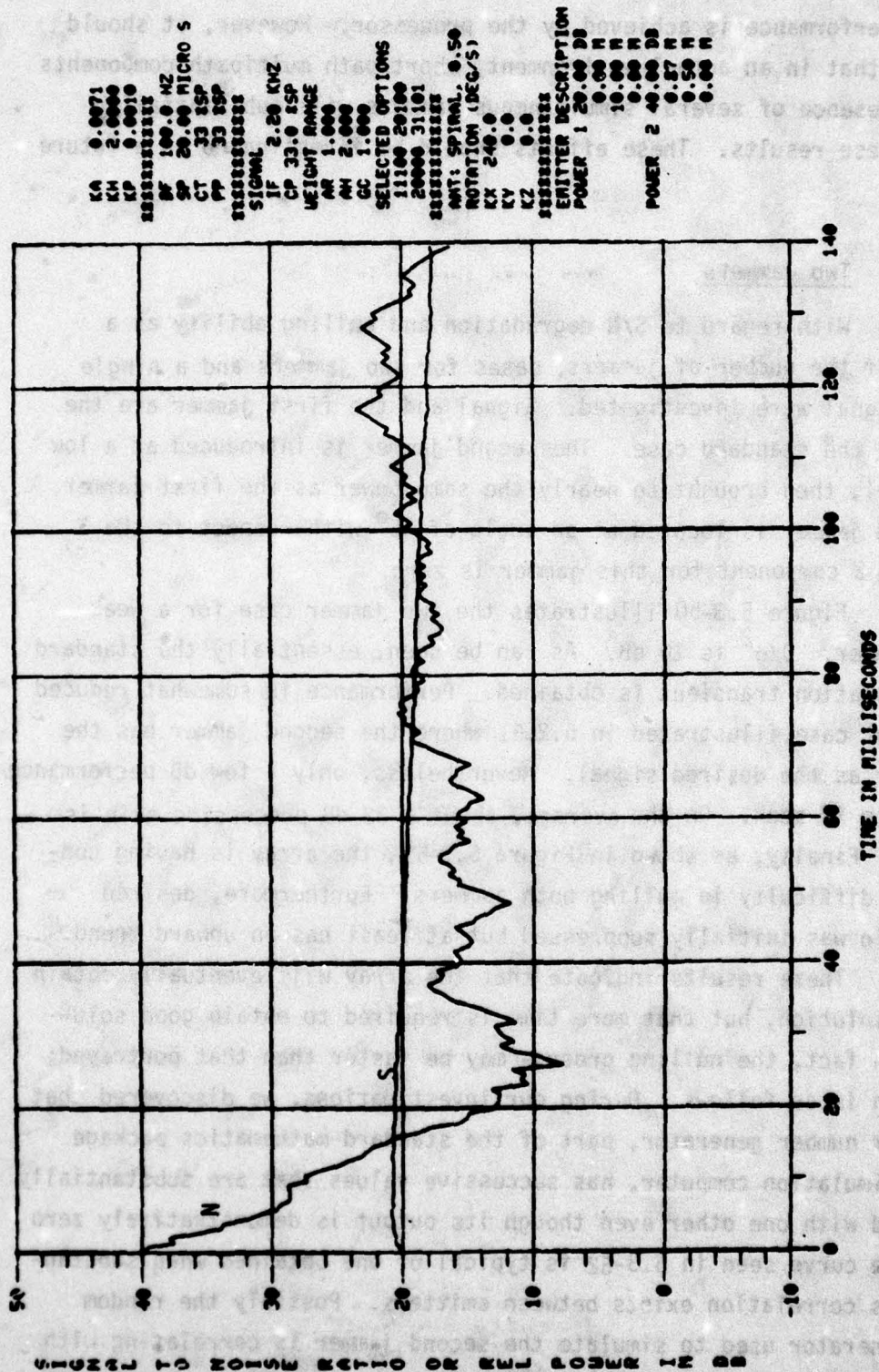


Figure 5.3-49. Adaptation Transients for Array Rotation at 240°/Sec

marginal performance is achieved by the processor. However, it should be noted, that in an actual environment, short path multipath components and the presence of several simultaneous jammers will substantially degrade these results. These effects should be investigated in a future study.

5.3.9 Two Jammers

With regard to S/N degradation and nulling ability as a function of the number of jammers, cases for two jammers and a single desired signal were investigated. Signal and the first jammer are the same as in the standard case. The second jammer is introduced at a low power level, then brought to nearly the same power as the first jammer. The second jammer is located at an angle of 45° with respect to the X axis. The Z component for this jammer is zero.

Figure 5.3-50 illustrates the two jammer case for a weak second jammer. J/σ^2 is 10 dB. As can be seen, essentially the standard value adaptation transient is obtained. Performance is somewhat reduced in the next case illustrated in 5.2-51 where the second jammer has the same power as the desired signal. Nevertheless, only a few dB performance degradation is seen. On the average, about a 32 dB processing gain is obtained. Finally, as shown in Figure 5.3-52, the array is having considerable difficulty in nulling both jammers. Furthermore, desired signal gain was initially suppressed but at least has an upward trend.

These results indicate that the array will eventually obtain a useful solution, but that more time is required to obtain good solutions. In fact, the nulling process may be faster than that portrayed; the reason is as follows. During our investigations, we discovered that the random number generator, part of the standard mathematics package for the simulation computer, has successive values that are substantially correlated with one other even though its output is demonstratively zero mean. The curve seen in 5.3-52 is typical of one obtained when substantial cross correlation exists between emitters. Possibly the random number generator used to simulate the second jammer is correlating with array thermal noise parameters also generated by that same subroutine.

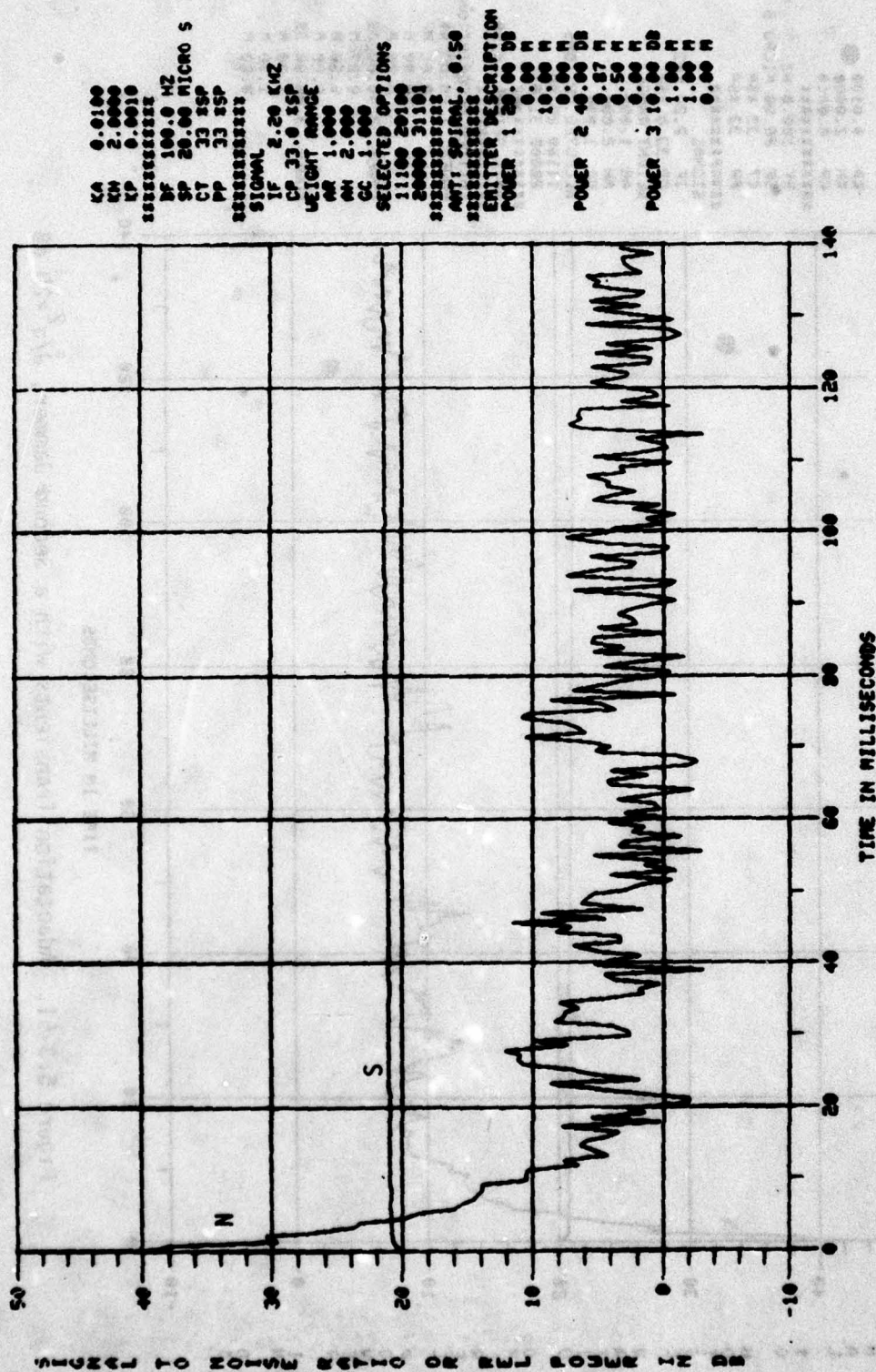


Figure 5.3-50. Adaptation with a Second Jammer, $J/\sigma^2=10$ dB

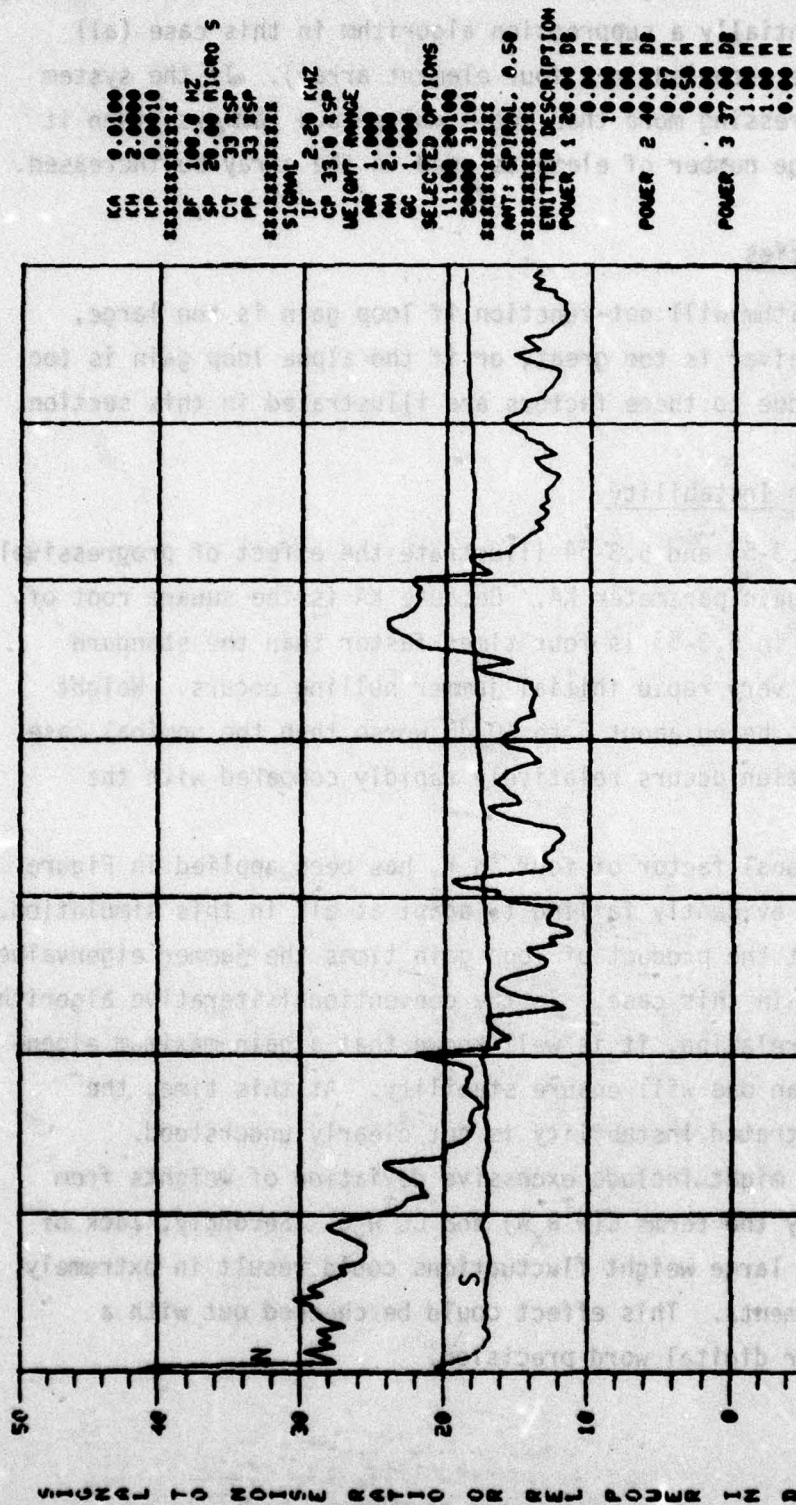


Figure 5.3-52. Adaptation Transients with a Second Jammer, $J/\sigma^2 = 37$ dB

The case for three jammers was not run because a S/N maximizing array reverts to essentially a suppression algorithm in this case (all degrees of freedom are occupied in a four element array). If the system scenario requires addressing more than two simultaneous jammers, then it is recommended that the number of elements used in the array be increased.

5.3.10 Instabilities

The algorithm will not function if loop gain is too large, time delay in the receiver is too great, or if the alpha loop gain is too high. Instabilities due to these factors are illustrated in this section.

5.3.10.1 Array Gain Instability

Figures 5.3-53 and 5.3-54 illustrate the effect of progressively increasing the array gain parameter K_A . Because K_A is the square root of the gain K_1 , the case in 5.3-53 is four times faster than the standard case. Observe that a very rapid initial jammer nulling occurs. Weight jitter is substantial, being about 6 to 10 dB worse than the nominal case. Desired signal adaptation occurs relatively rapidly compared with the standard case.

An additional factor of four in K_1 has been applied in Figure 5.3-54. The array is evidently failing to adapt at all in this simulation. It is interesting that the product of loop gain times the jammer eigenvalue is much less than one in this case. In the conventional iterative algorithm using a direct RF correlation, it is well known that a gain-maximum eigenvalue product less than one will ensure stability. At this time, the reason for the demonstrated instability is not clearly understood. Several possibilities might include excessive deviation of weights from their static values by the terms $C(W^T R_X W)$ and $CC^T R_X C$. Secondly, lack of precision due to very large weight fluctuations could result in extremely poor gradient measurements. This effect could be checked out with a machine having greater digital word precision.

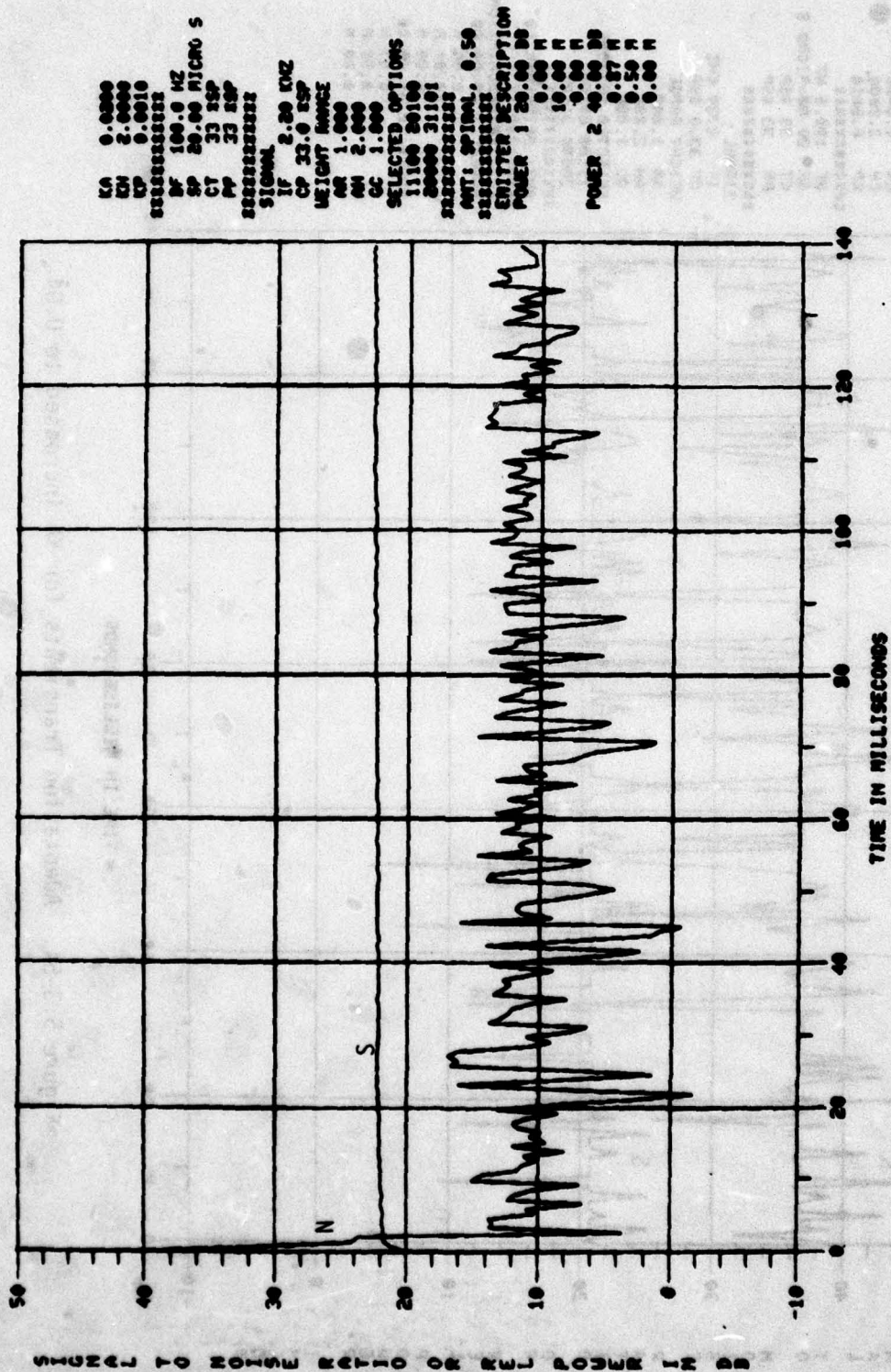


Figure 5.3-53. Adaptation Transients for KA Increased to 0.02

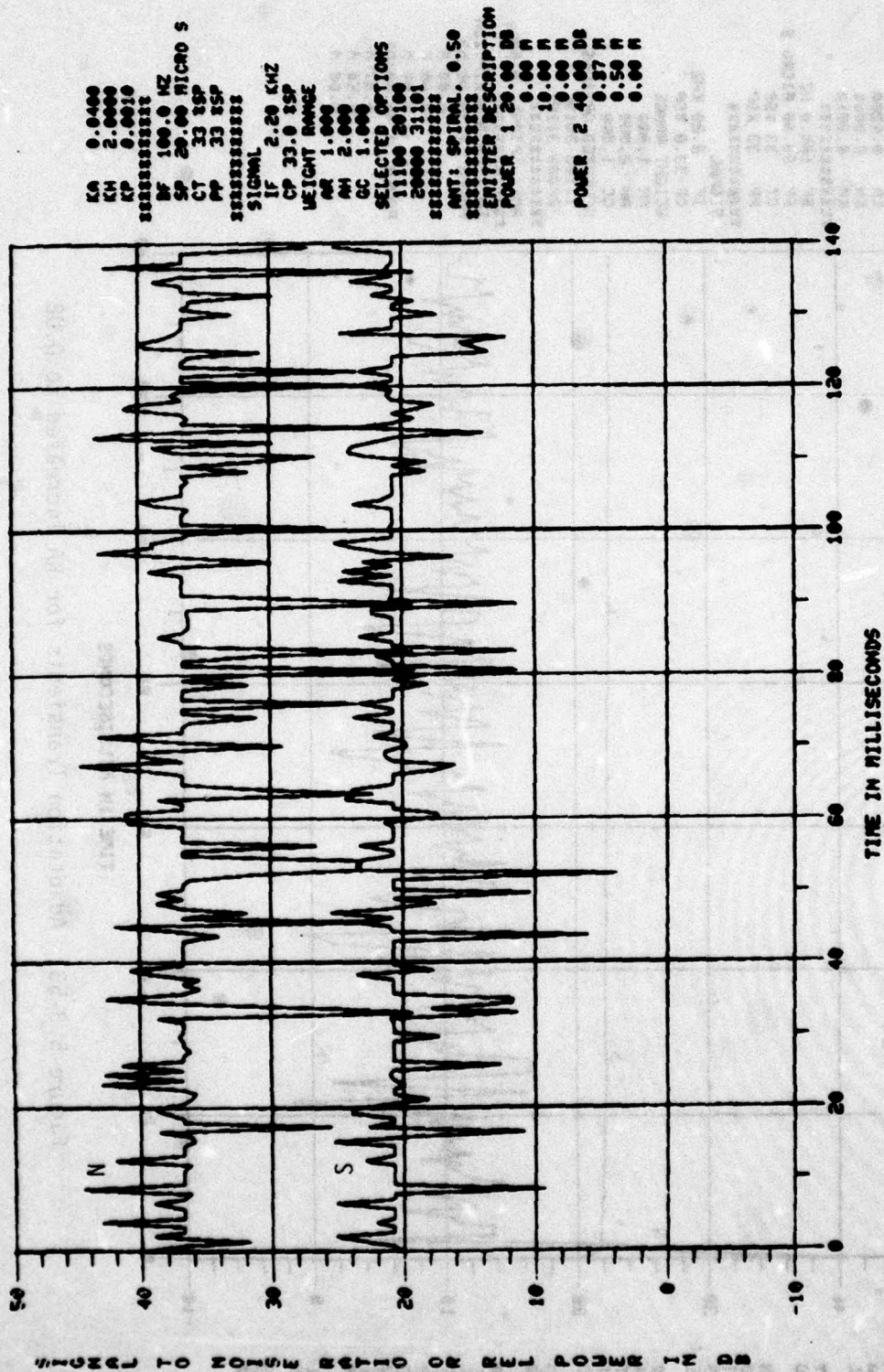


Figure 5.3-54. Adaptation Transients for KA Increased to 0.04

We can evaluate the gain-jammer eigenvalue product and get

$$K\lambda_J = 2(KA)^2(KP)(KW)\lambda_J \approx 0.096 . \quad 5-15$$

As can be seen, this product is roughly an order of magnitude less than that which should cause instability using the conventional analysis.

Furthermore, a simulation for the same parameters causing the instability observed in 5.3-54 is repeated in 5.3-55 except that KP is reduced by an order of magnitude. This, of course, reduces the circuit gain by another order of magnitude and thus should result in a stable algorithm. As can be seen from the simulation runs, such is not the case. A theoretical investigation needs to be directed toward understanding this effect.

5.3.10.2 Receiver Time Delay

Increasing the delay between the time of weight adjustment and calculation of the new weight difference is well known to produce instability in iterative circuits. This effect is illustrated in Figures 5.3-56 through 5.3-60. The standard case is run except that receiver time delay is increased from the standard one perturbational period to many perturbational periods. Figure 5.3-56 is for the case of receiver delay equal to two perturbational periods. This delay increases by a factor of two each case until Figure 5.3-60 where the delay is 64 periods. Inspection of the adaptation transients obtained indicates a progressive deterioration of stability as the delay is increased. It is interesting that the loop gain-jammer eigenvalue product times the receiver delay is in the order of unity when the instability occurs (0.3). Whether or not this is helpful in understanding the previous instability remains to be seen.

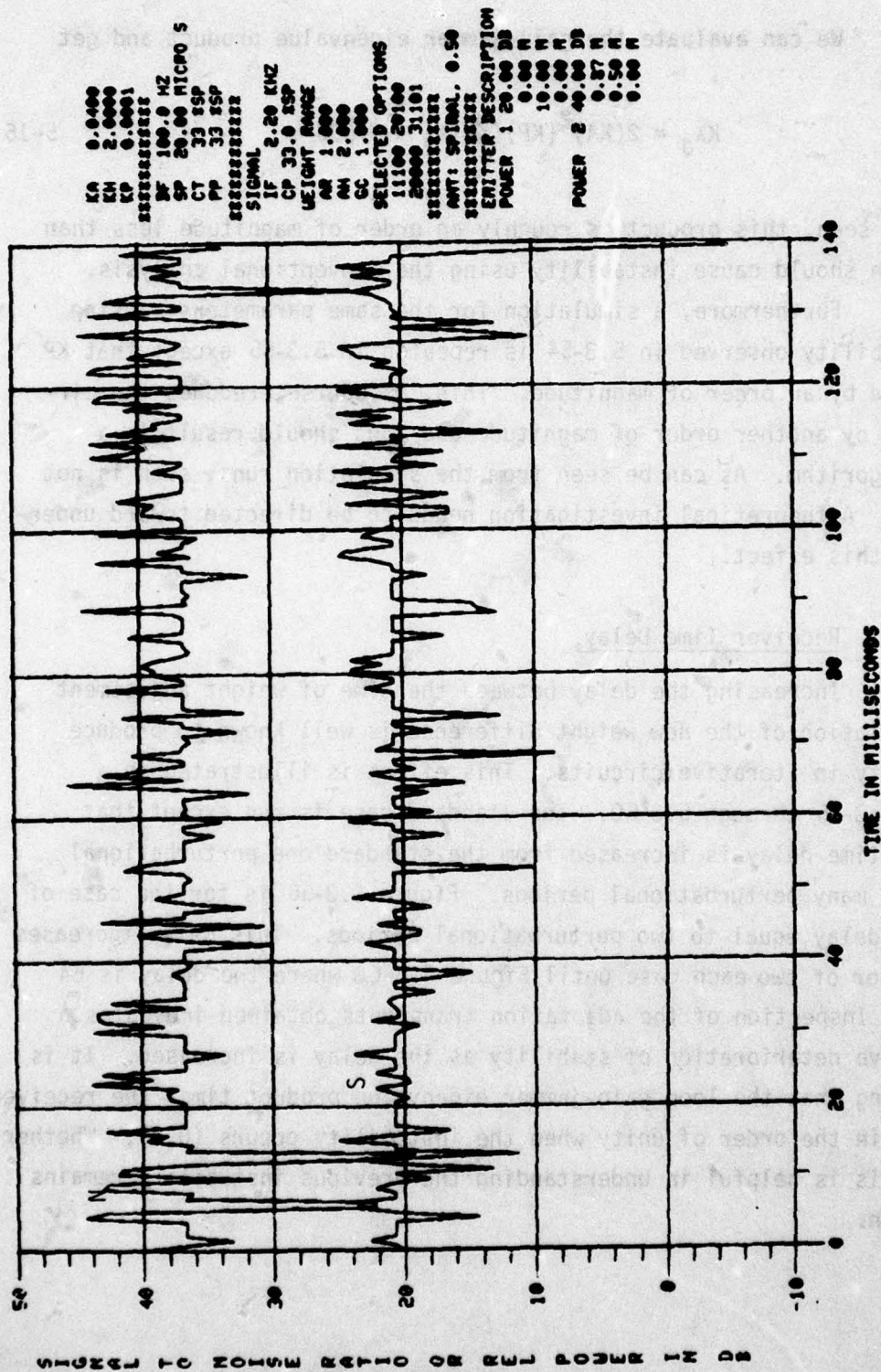


Figure 5.3-55. Adaptation Transients for KA Increased to 0.04 and KP Decreased to 0.0001

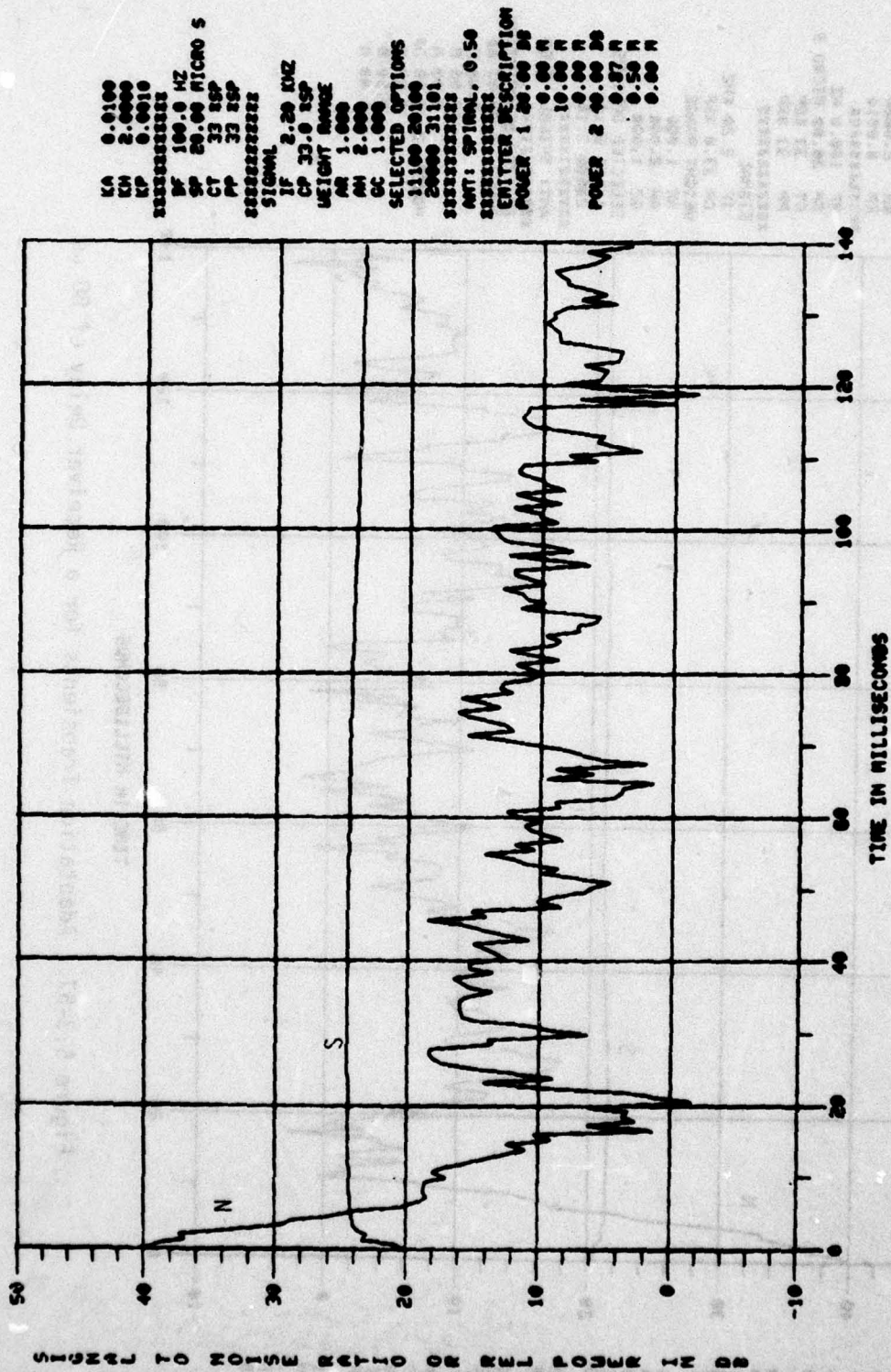


Figure 5.3-56. Adaptation Transients for a Receiver Delay of 40 μ s

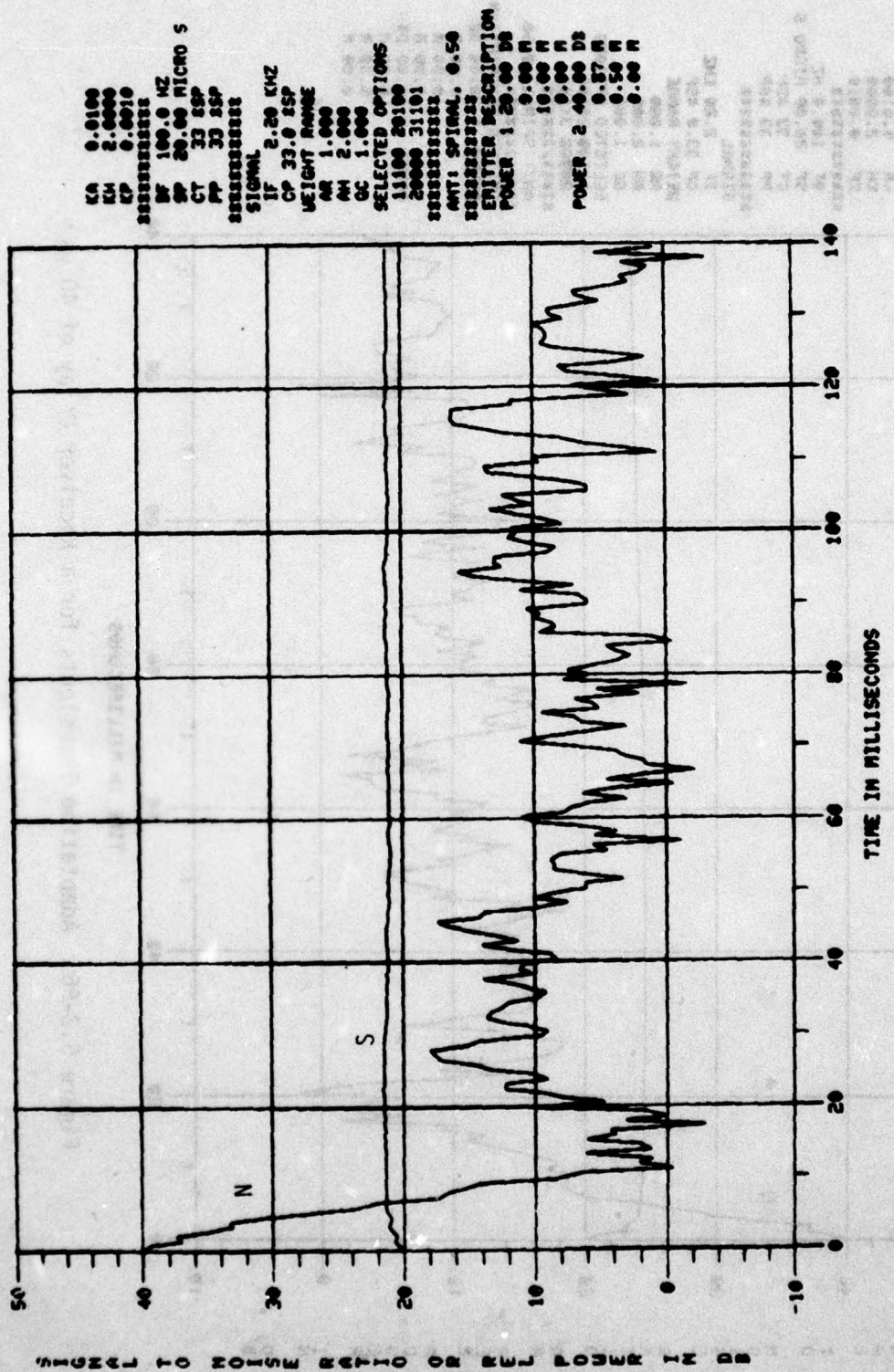


Figure 5.3-57. Adaptation Transients for a Receiver Delay of 80 μ s

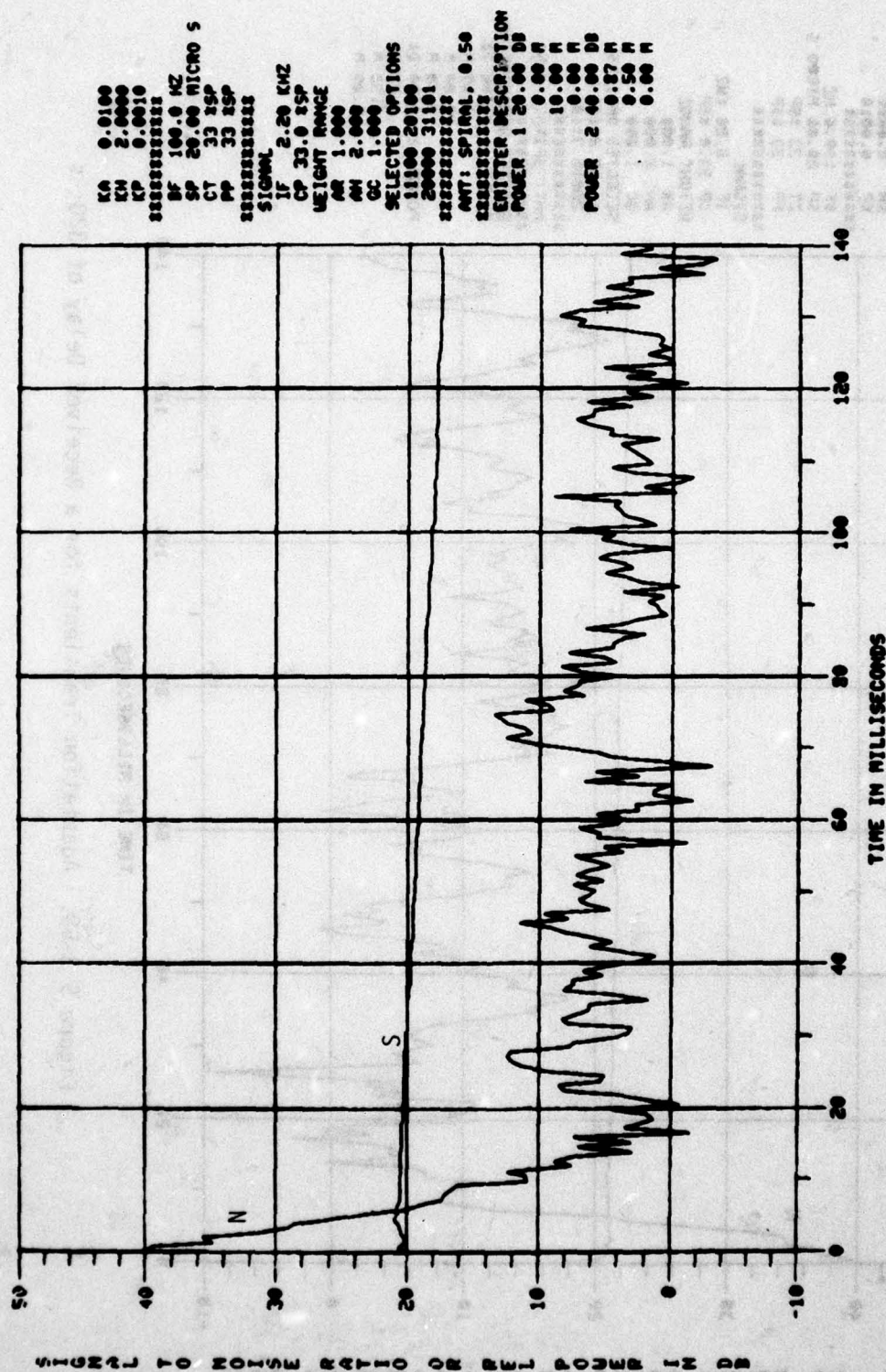


Figure 5.3-58. Adaptation Transients for a Receiver Delay of 160 μ s

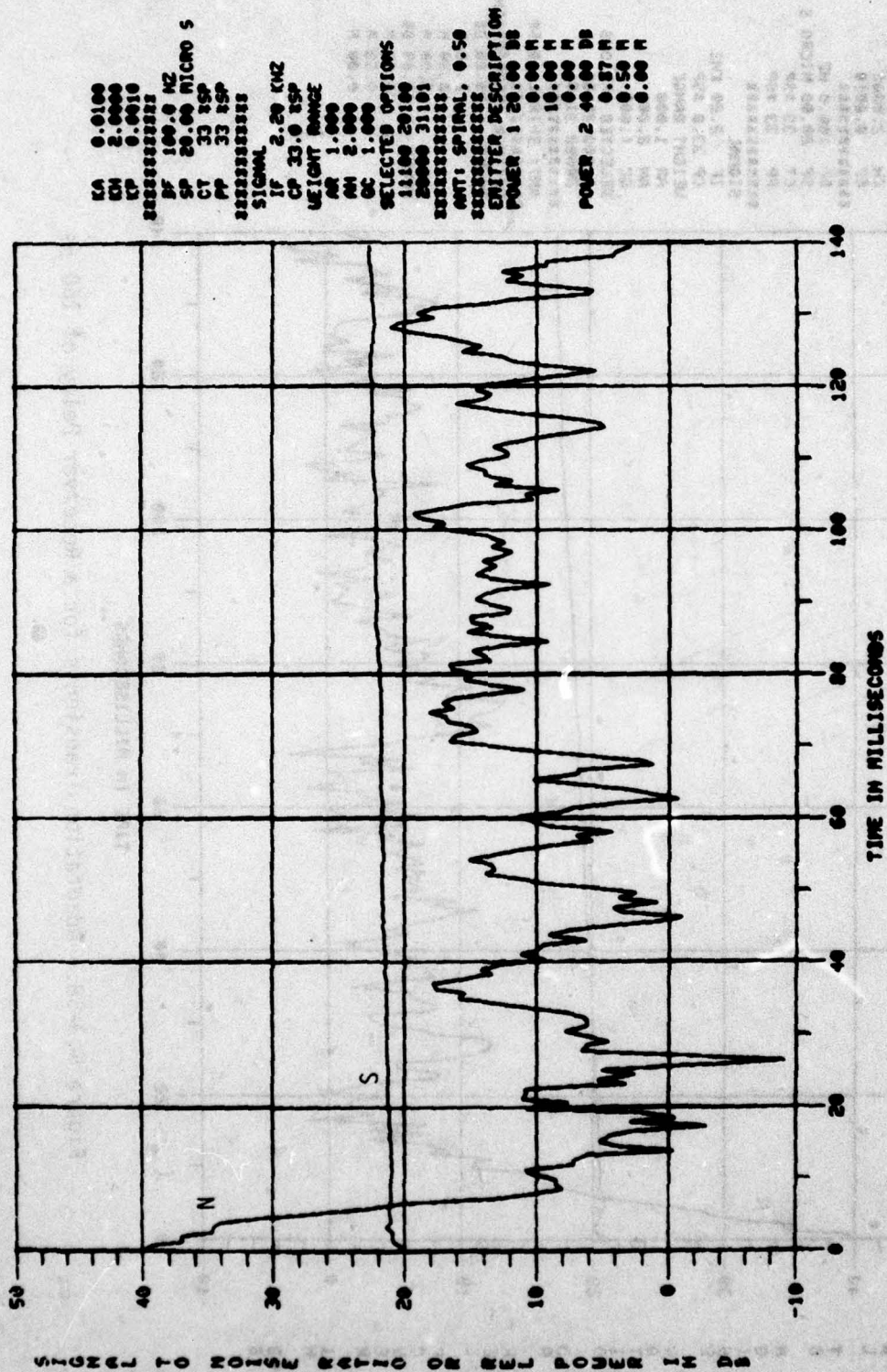


Figure 5.3-59. Adaptation Transients for a Receiver Delay of 320 μ s

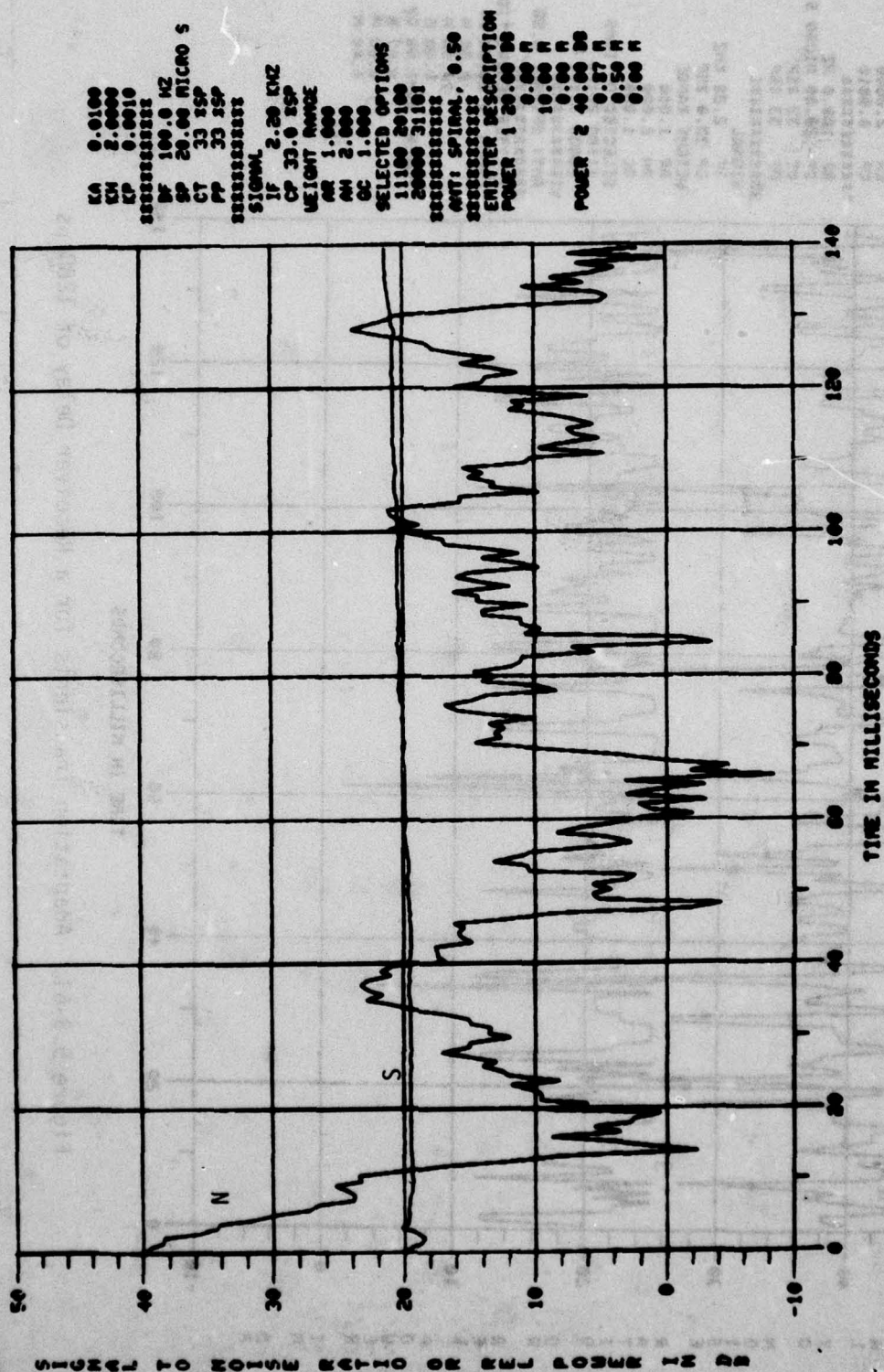
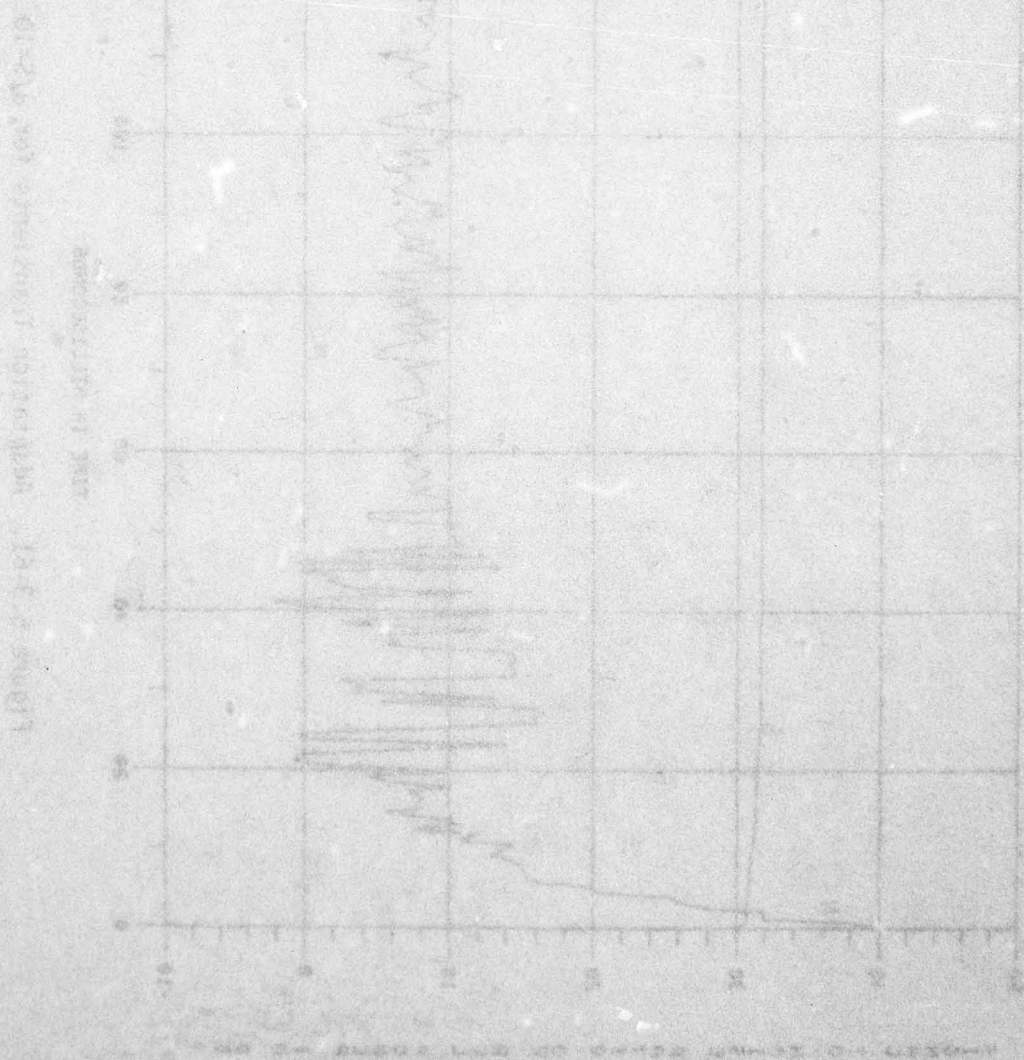


Figure 5.3-60. Adaptation Transients for a Receiver Delay of 640 μ s

5.3.11 Signal Power

Input J/S is varied ± 10 dB in the following two runs. In each case the noise power transient is essentially the same. Note that in Figure 5.3-61 for the case of 10 dB J/S, substantial desired signal adaptation occurs in the 0 to 20 ms time period, as expected from loop gain and signal eigenvalue considerations. Note that the initial fast rise in desired signal power is a consequence of the jammer null indirectly placing part of a pattern lobe in the desired signal direction.



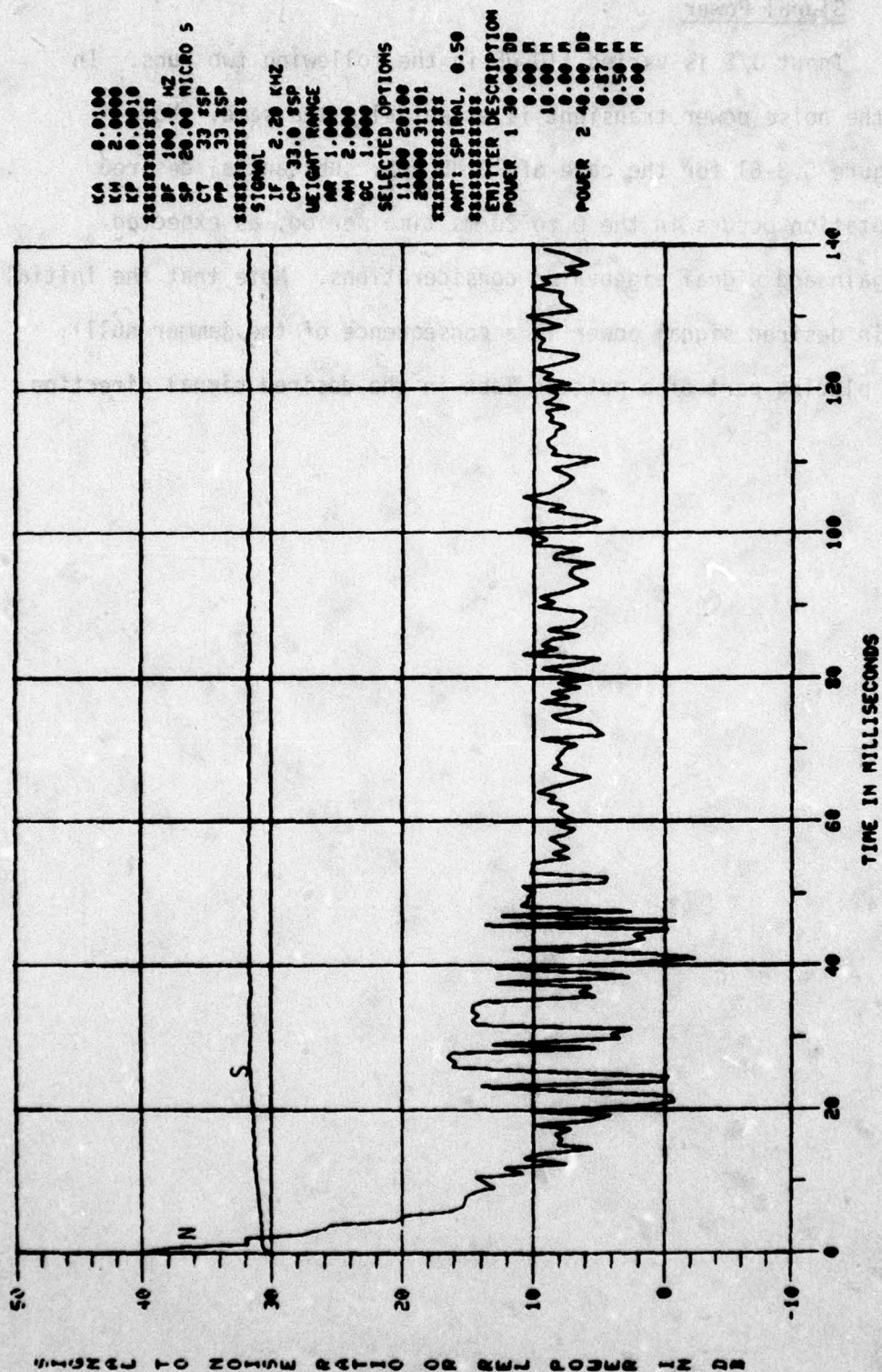


Figure 5.3-61. Adaptation Transients for J/S=10 dB

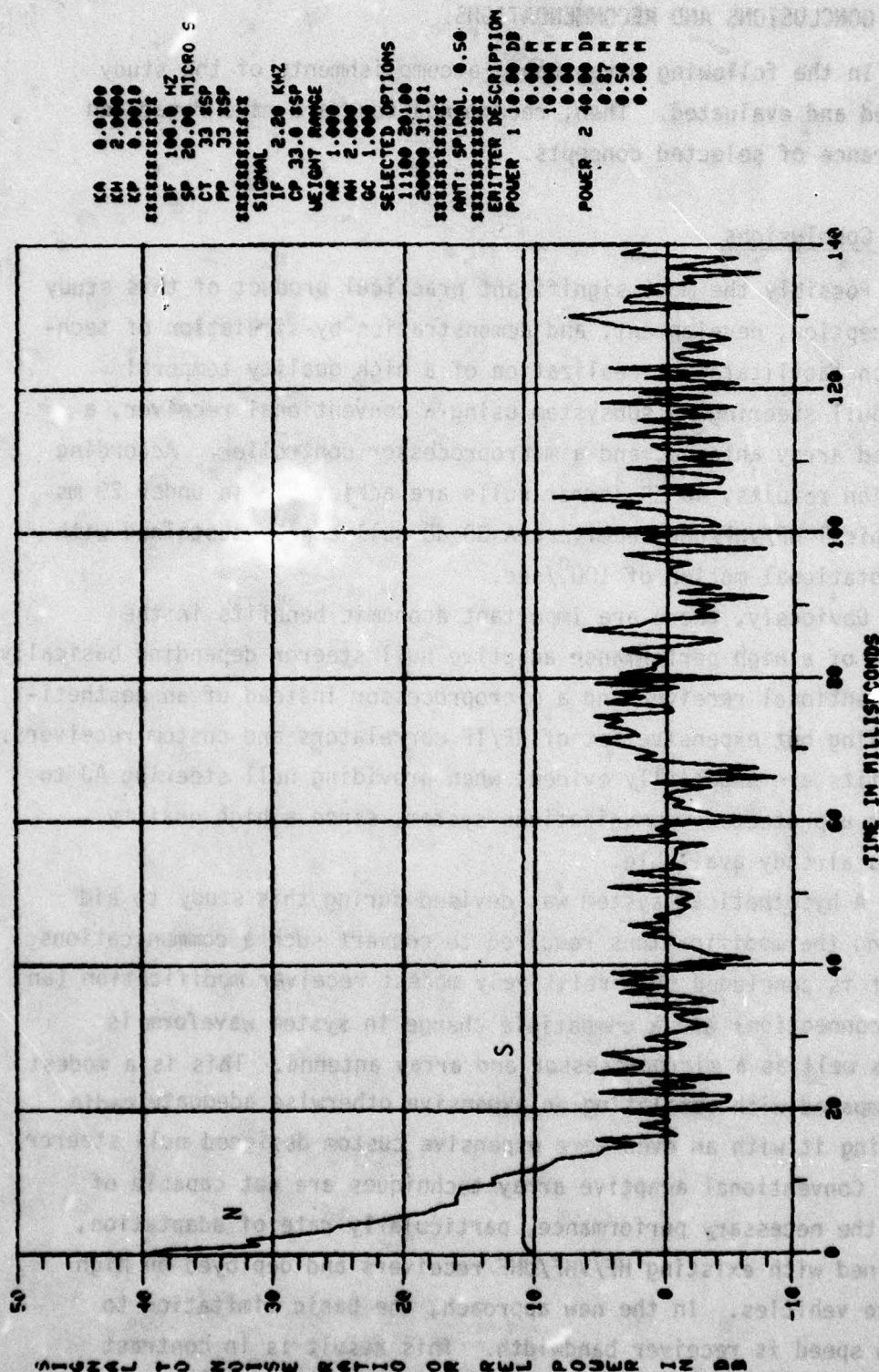


Figure 5.3-62. Adaptation Transients for J/S=30 dB

6.0 CONCLUSIONS AND RECOMMENDATIONS

In the following paragraphs, accomplishments of the study are recalled and evaluated. Then, recommendations are made regarding the furtherance of selected concepts.

6.1 Conclusions

Possibly the most significant practical product of this study is the conception, development, and demonstration-by-simulation of techniques which facilitate the realization of a high quality temporal reference null steering AJ subsystem using a conventional receiver, a small phased array antenna, and a microprocessor controller. According to simulation results, 40 dB jammer nulls are achievable in under 20 ms using a typical HF/VHF/UHF receiver; A 30 dB null can be sustained with an array rotational motion of $100^{\circ}/\text{sec}$.

Obviously, there are important economic benefits in the achievement of a high performance adaptive null steerer depending basically upon a conventional receiver and a microprocessor instead of an aesthetically pleasing but expensive set of RF/IF correlators and custom receivers. These benefits are especially evident when providing null steering AJ to an existing unprotected communications system, since a high quality receiver is already available.

A hypothetical system was devised during this study to aid in assessing the modifications required to convert such a communications system. It is concluded that relatively modest receiver modification (an IF output connection) and a compatible change in system waveform is required as well as a microprocessor and array antenna. This is a modest penalty compared with obsoleting an expensive otherwise adequate radio and replacing it with an even more expensive custom designed null steerer.

Conventional adaptive array techniques are not capable of providing the necessary performance, particularly rate of adaptation, when combined with existing HF/VHF/UHF receivers and deployed on high performance vehicles. In the new approach, the basic limitation to adaptation speed is receiver bandwidth. This result is in contrast

to a conventional temporal reference algorithm which is limited by the data bandwidth of the desired signal or, in more severe cases, by the bandwidth of the desired signal carrier tracking loop. Generally one or more orders of magnitude faster adaptation can be achieved by the new approach.

A primary factor in realizing this speed was the development of the Correlation Discriminant Operator, or Signal Recognizer. Since the signal recognizer is dependent upon noise decorrelation rather than noise filtering, an unbiased though sometimes noisy measurement of desired signal power can be made on an "instantaneous" basis.

An additional primary factor was the conception and development of algorithms both in differential equation form and in novel perturbational form which eliminates the conventional complex and slow "Performance Measure" device. Paralleling this effort was a development of the modified PSF algorithm. When used with a signal recognizer, the PSF achieves optimum broadband S/N results. Additionally, this algorithm avoids the "weight AGC" effect and the related zero weight solution for an absent desired signal that characterizes the LMS algorithm. In addition a generalized version of the PSF was derived which accommodates the case of multiple simultaneous desired signals; this optimal signal to noise ratio maximizing algorithm is expected to be useful for conferencing system users.

Although it is not directly related to the original study objectives, the Random Search algorithm analysis result showing equivalence with a normalized gradient algorithm is significant in that fundamental understanding and hence better design of that class of algorithm is enabled.

6.2 Recommendations

In the course of realizing the original study objectives, more practical and theoretical avenues were opened than closed. These avenues lead to three areas of consideration, the first of which follows directly from the original objectives.

On the basis of the simulations reported in Chapter 5.0, it appears feasible and desirable to achieve in hardware a complete temporal reference perturbational null steering AJ subsystem integrally incorporating a "to be selected" communications receiver. A low level refined analytical study should be conducted in parallel with the experimental effort. These topics are expanded in Section 6.2.1.

The second area of consideration encompasses topics closely related to the original objectives, which although of theoretical and practical interest, are not central to the success of an experimental program. These subjects are listed in 6.2.2.

Topics in the third area are substantially separate from the other two in that they do not directly relate to the original Study objectives. Included here is the Random Search/Normalized Gradient algorithm equivalence, the dual phase shifter weighting technique, the error phase tolerant algorithm, and the self orthogonalizing algorithm.

6.2.1 Direct Applications

By rough estimate, the utilization of a communications set receiver as an integral part of a null steering AJ subsystem, whether used in a new system design or for upgrading an existing system, should enable an adaptive processor to be realized at a cost savings of at least an order of magnitude with respect to the conventional direct correlation approach. According to the simulation results presented in Chapter 5.0, the performance of such an algorithm can be nearly as good as the direct (RF/IF) correlation technique in all but the most critical applications. Consequently, an experimental program is recommended utilizing a government-selected communications set in conjunction with the algorithms and microprocessor circuits developed in this Study. Useful experimental data could be expeditiously obtained for a 4-element array, two jammers and a single desired signal.

A preliminary design and construction phase should precede the experimental work, the goal being selection and assembly of the array antenna, weighting circuits, A/D converters, sample and hold

AD-A068 890

HARRIS CORP MELBOURNE FLA

F/G 17/4

APPLICATION OF A CORRELATION DISCRIMINANT OPERATOR TO PERTURBAT--ETC(U)

APR 79 6 P MARTIN

F30602-77-C-0073

NL

UNCLASSIFIED

RADC-TR-79-44

4 OF 4
ADA
068890



END
DATE
FILMED
8 79
DDC

circuits, and other hardware components. Additionally, final determination of algorithm parameters such as absolute gain, perturbation amplitude, perturbation functions, desired signal chip rate, etc. is required.

In parallel with the experimental effort, there should be a theoretical investigation refining the performance projections of this study. An important topic to be addressed is the effect of the selected receiver's bandpass filter characteristics upon the perturbational sequences and consequently, ability of the circuit to accurately compute the error surface gradient. Also detailed effects of multiple jammers and array motion should be determined. Output from this effort would be comprehensive performance measurements, detailed simulation data for comparison with the measured data, size, weight, and power determinations, and accurate costs.

6.2.2 Related Topics

Although these related topics could be addressed in a direct application analysis program like that just described, such an investigation is not required for a successful experimental program. Nevertheless, it is believed that these theoretical questions and suggestions for algorithm improvement are important and should be studied.

A multiple simultaneous desired signal version of the PSF algorithms was derived and presented in Section 3.1.3. This algorithm was designed to simultaneously optimize the response to several desired signals in a jamming environment such as might be seen during "conferencing," but time did not permit further development or simulation of this algorithm. In addition to adaptive simulations, an investigation should also address modem interface and acquisition techniques.

With regard to simulation data presented in Chapter 5.0 of this report continually updated and periodically updated weight transients should be identical in an expected value sense. Frequently, however, differences in the adaptation were seen for specific cases. Since only a few cases were examined, it is possible that upon averaging many cases,

equivalent performance will be seen. If not, the reason for the difference should be determined.

Regarding algorithm instabilities, application of conventional stability criteria did not predict the loop gain values at which the perturbational algorithm would become unstable. The reason for this should be determined, and if necessary, a new theory developed or old theory revised.

It is noted that the "DC" terms arising from the instantaneous array output power multiplier and the corresponding desired signal multiplier do not contribute to the algorithm operation. Furthermore, the presence of these terms increase precision requirements for the digital processing circuits. It is suggested that highpass filtering to eliminate these terms could either substantially improve performance or substantially reduce the cost and complexity of the microprocessor controller. Of particular interest here would be the FDMA perturbational approach discussed in 3.3.2.

6.2.3 New Directions

In these paragraphs, we address topics of considerable interest which do not directly relate to the objectives of this study, but which are highly relevant to adaptive array technology and promise to be worthy of additional analysis and development.

Random search optimization techniques are very important in adaptive array applications, particularly for large arrays due to their great economy in hardware, and thus, ultimate system cost, but the fact that these algorithms have been very difficult to analyze has hampered their development and utilization. In the course of this study, we were able to determine the basis of the random search algorithm. Specifically, it was shown that in an expected value sense, it is equivalent to a normalized gradient algorithm. This equivalence can now be used to deterministically design better search algorithms and to place empirically based state-of-the-art procedures on firm theoretical ground. It is recommended that a study be undertaken to consolidate the new theory

with state-of-the-art technology and to further the fundamental understanding of random search algorithms.

The novel dual phase shifter weight presented in Chapter 4.0 has the promise of improving adaptive array performance while reducing the need for RF amplification. Theoretical and physical properties of this technique should be investigated including various realizations such as switched line, varactor-hybrid, and phase lock loop approaches (inherent frequency conversion in addition to weighting). Furthermore, techniques for obtaining coordinate transformations between complex variables and phase domain variables should be studied.

Although the error phase tolerant algorithm details were not included in this report, this approach is worthy of additional study because it removes the restriction of controlled phase in the algorithms' error path as required for stability of conventional algorithms such as the LMS and PSF.

Finally, a self orthogonalizing algorithm was devised during this study but was not pursued because initial analysis indicated it was excessively noisy. Nevertheless, different circuit configurations or different applications may make this a practical technique. Its primary advantage is that the adaptation transient for all emitters can be made the same. In this sense, it is similar to recursive estimation techniques. Of course, the value in having all emitters display the same adaptation rate is that convergence to desired signal optimization results need not be orders of magnitude longer than jammer nulling effects.

APPENDIX A

GENERAL CALCULATION OF PERTURBATIONAL ALGORITHM SIGNAL RECOGNIZER OUTPUT

The purpose of this appendix is to provide an analysis of the output of a signal recognizer circuit when used in a perturbational algorithm having very rapid perturbations. In this case, signal recognizer delay may be many perturbational periods.

Reference to Figure A-1 illustrates the circuit diagram being analyzed. Antenna inputs $x_i(t)$ are weighted by a quasi-static weight W_i plus a perturbational sequence $c_i(t)$. The receiver output, $y(t)$, is therefore

$$y(t) = [W + C(t)]^T x(t) \quad A-1$$

where we have used vector notation to sum the weighted inputs. For comparison purposes, we calculate the instantaneous power output obtained by the left-hand multiplier and get

$$P_x(t) = W^T x x^T W + C^T x x^T C + 2C^T x x^T W \quad A-2$$

where we have squared and expanded (A-1). The correlation discriminant operator (CDO) output is

$$r(t) = [W + C(t - \tau)]^T [S(t) + n(t)] \quad A-3$$

where τ is the time delay of the CDO. Recall that the signal waveform is "not delayed" by virtue of the structure of the CDO. The instantaneous

power $P_S(t)$ is obtained by multiplying (A-3) by (A-1). We obtain

$$P_S(t) = [W+C(t-\tau)]^T [S+n][S+n]^T [W+C(t)]. \quad A-4$$

Expanding (A-4) gives

$$P_S(t) = [W+C(t-\tau)]^T [SS^T + nS^T + Sn^T + nn^T] [W+C(t)]. \quad A-5$$

From CDO theory, we know that the only matrix term in (A-5) having a non-zero expected value is the matrix SS^T . The other terms contribute noise which ultimately causes weight jitter but they do not contribute a bias. We will lump all of these other terms together into one term $N(t)$, expand and get

$$P_S(t) = W^T SS^T W + C^T(t-\tau) SS^T C(t) + C^T(t-\tau) SS^T W + C^T(t) SS^T W + N. \quad A-6$$

First, we consider the case when the perturbational sequence $C(t)$ is periodic (for example, Walsh functions). In this case, if τ is equal to one or more complete periods of the perturbational sequence, then we have

$$C(t-\tau) = C(t). \quad A-7$$

Therefore, in this case we obtain the same value for P_S as in the case where τ is negligibly small. We get

$$P_S(t) = W^T SS^T W + C^T SS^T C + 2C^T SS^T W + N. \quad A-8$$

Following the procedure of Chapter 3.0 we calculate the gradient of the signal power by multiplying P_s by the perturbational sequence vector C , take expected values and obtain

$$E\{CP_s\} = 2E\{CC^T SS^T W\} = 2R_s W \quad 4-9$$

where we have used the fact C has the properties delineated in Chapter 3.0; that is, zero mean, simple orthogonality and triplet orthogonality. Recall that $E\{CC^T\} = I$.

Consider now that the sequence $C(t)$ is random or pseudo random. Evidently, we must multiply P_s by either $C(t)$ or $C(t-\tau)$ in order to obtain a useful correlation. We choose $C(t)$, perform the indicated multiplication and take expected values getting

$$E\{CP_s\} = E\{C(t)C^T(t-\tau)SS^T W\} + E\{C(t)C^T(t)SS^T W\} = R_s W \quad 4-10$$

where we recognize that $E\{C(t)C^T(t-\tau)\} \approx 0$ if $\tau > \tau_{\text{chip}}$. We observe that the gradient obtained is half that for the cases when $C(t)$ is periodic or the time delay is negligibly small.

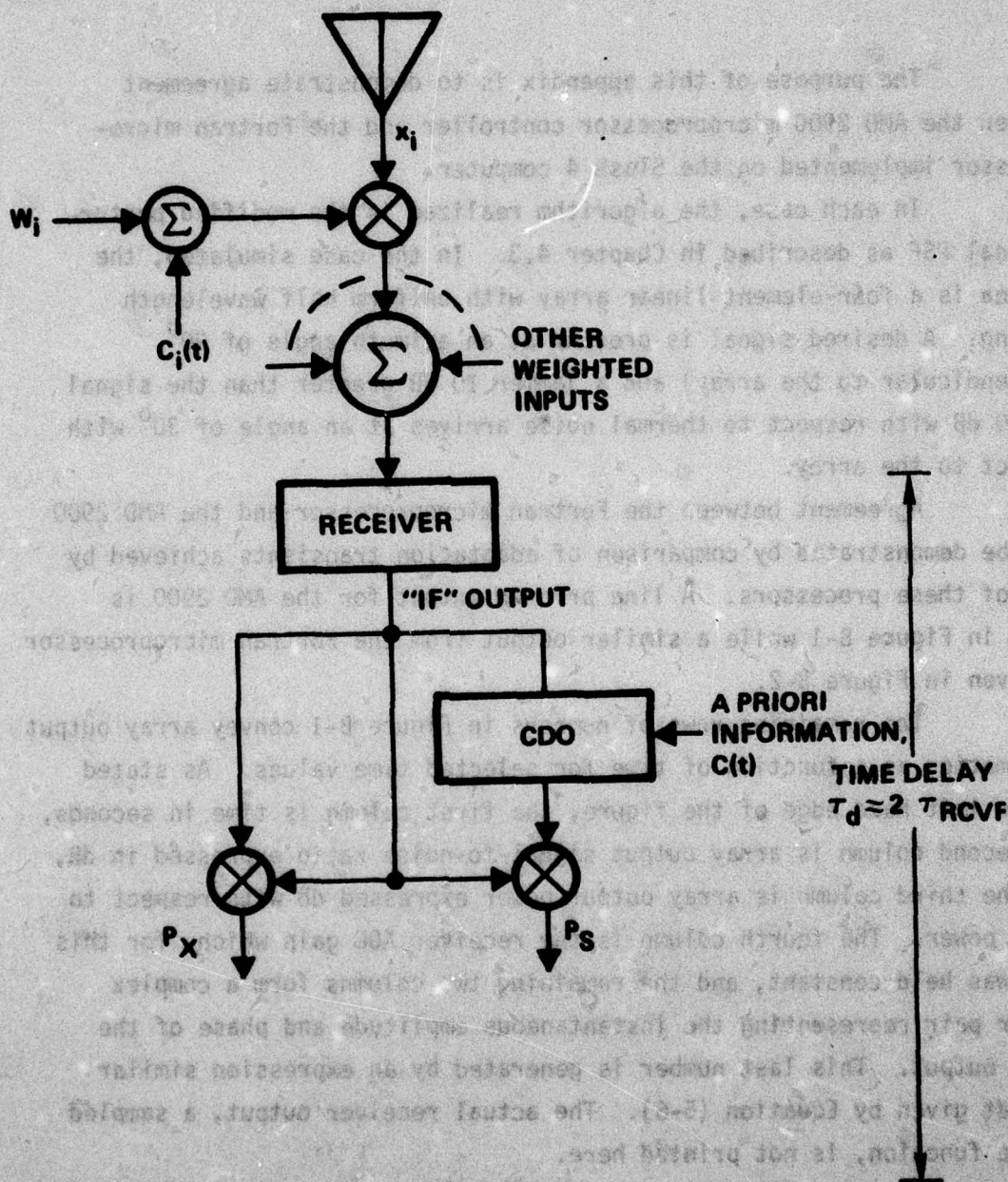


Figure A-1. Perturbational Algorithm Using A Correlation Discriminant Operator

APPENDIX B

AGREEMENT OF COMPUTER SIMULATIONS

The purpose of this appendix is to demonstrate agreement between the AMD 2900 microprocessor controller and the Fortran microprocessor implemented on the Slash 4 computer.

In each case, the algorithm realized is the modified perturbational PSF as described in Chapter 4.3. In the case simulated, the antenna is a four-element linear array with uniform half wavelength spacing. A desired signal is present at an azimuth angle of 90° (perpendicular to the array) and a jammer 10 dB greater than the signal and 40 dB with respect to thermal noise arrives at an angle of 30° with respect to the array.

Agreement between the Fortran microprocessor and the AMD 2900 will be demonstrated by comparison of adaptation transients achieved by each of these processors. A line printer output for the AMD 2900 is given in Figure B-1 while a similar output from the Fortran microprocessor is given in Figure B-2.

The remaining rows of numbers in Figure B-1 convey array output information as a function of time for selected time values. As stated at the left hand edge of the figure, the first column is time in seconds, the second column is array output signal-to-noise ratio expressed in dB, and the third column is array output power expressed dB with respect to input power. The fourth column is the receiver AGC gain which, for this run, was held constant, and the remaining two columns form a complex number pair representing the instantaneous amplitude and phase of the array output. This last number is generated by an expression similar to that given by Equation (5-6). The actual receiver output, a sampled cosine function, is not printed here.

At time zero, array output signal-to-noise ratio is -10 dB as expected. Normalized output power is 16.33 dB due to the fact that a single fixed weight having an omni antenna pattern is "on" at time zero. The nominal value of this weight is 0.0625. The complex

representation of the array output voltage is $(3.28+j6.22)$. Comparison with Figure B-2 reveals that both simulations have the same initial conditions.

For convenience, let us examine array output quantities at a time corresponding to the value at the end of the first block of numbers in Figure B-2. Time for this output is 0.01216 seconds. Comparison of the output values shows that agreement to roughly three places is obtained. Selection of other time values in this first block will demonstrate the same agreement precision.

When this data was obtained, the Fortran microprocessor was run with full precision while the AMD 2900 used 16-bits. Agreement past three places is not expected since the weight perturbation value (and thus gradient precision) is 6 bits.

The slightly different format between the two outputs is due to the fact that the Fortran microprocessor was programmed to print weight values achieved after a given number of iterations. The weights, given as complex number pairs and arranged vertically according to antenna element number, are shown in Figure B-2. Although corresponding numbers for the AMD 2900 are not given, agreement corresponding to about 6 bits precision is obtained.

The last value of time shown in Figure B-2 is 24.96 milliseconds. Due to precision differences in timekeeping, the AMD 2900 corresponding value is slightly less, being 24.9597 milliseconds. At this time, good agreement in all parameters is obtained.

These transients were continued until 76 milliseconds elapsed time with agreement dropping to about $2\frac{1}{2}$ places. Notice, however, that at this time, the jammer has been well suppressed, and according to the array output power column, desired signal gain is near maximum and is increasing slowly. Thus, the numbers given here represent one of the most interesting portions of the transient.

On the basis of the agreement shown here, it is concluded that the Fortran microprocessor and the AMD 2900 microprocessor produce essentially equivalent adaptation transients.

TIME, S/N	DR, POWER, AGC, COM, AV:	16.33152580	1.00000000	3.28449202	-6.11158	6
TIME, S/N	DB, POWER, AGC, COM, AV:	16.53802490	1.00000000	1.12527871	-5.39225	4
TIME, S/N	DR, POWER, AGC, COM, AV:	15.15776634	1.00000000	-0.68678701	-4.92515	1
TIME, S/N	DB, POWER, AGC, COM, AV:	15.68816948	1.00000000	3.36997080	-5.85751	5
TIME, S/N	DR, POWER, AGC, COM, AV:	15.23049259	1.00000000	-1.56251752	-4.19036	1
TIME, S/N	DB, POWER, AGC, COM, AV:	15.05717659	1.00000000	4.57837582	-4.97836	3
TIME, S/N	DR, POWER, AGC, COM, AV:	15.12758398	1.00000000	3.32203694	-4.33993	4
TIME, S/N	DB, POWER, AGC, COM, AV:	12.77694321	1.00000000	-1.78685856	-2.32043	6
TIME, S/N	DR, POWER, AGC, COM, AV:	12.5774250	1.00000000	-2.27085114	-0.55550	8
TIME, S/N	DB, POWER, AGC, COM, AV:	11.55967712	1.00000000	-2.81119394	-0.83905	7
TIME, S/N	DR, POWER, AGC, COM, AV:	11.08794117	1.00000000	-2.94544649	-0.61649	6
TIME, S/N	DB, POWER, AGC, COM, AV:	10.78057194	1.00000000	-2.51070023	-2.65579	0
TIME, S/N	DR, POWER, AGC, COM, AV:	10.42159557	1.00000000	2.39931440	-2.64970	0
TIME, S/N	DB, POWER, AGC, COM, AV:	10.58601665	1.00000000	-2.90380263	-0.17856	7
TIME, S/N	DR, POWER, AGC, COM, AV:	10.46936589	1.00000000	-3.05699420	-2.49000	4
TIME, S/N	DB, POWER, AGC, COM, AV:	10.43539810	1.00000000	-2.54542451	-0.54616	8
TIME, S/N	DR, POWER, AGC, COM, AV:	10.48737335	1.00000000	3.22189689	-2.48145	6
TIME, S/N	DB, POWER, AGC, COM, AV:	10.45132446	1.00000000	3.17859721	-2.16606	2
TIME, S/N	DR, POWER, AGC, COM, AV:	10.02986908	1.00000000	-2.30497408	0.31737	4
TIME, S/N	DB, POWER, AGC, COM, AV:	9.92837334	1.00000000	3.41056991	-1.92504	0
TIME, S/N	DR, POWER, AGC, COM, AV:	9.94532585	1.00000000	-2.04839730	0.50836	7
TIME, S/N	DB, POWER, AGC, COM, AV:	9.78861618	1.00000000	3.42152572	-1.83372	4
TIME, S/N	DR, POWER, AGC, COM, AV:	9.84254932	1.00000000	-1.84332478	0.60043	0
TIME, S/N	DB, POWER, AGC, COM, AV:	9.90150738	1.00000000	3.55148029	-1.89514	6
TIME, S/N	DR, POWER, AGC, COM, AV:	9.93894196	1.00000000	-1.60487306	0.49392	7
TIME, S/N	DB, POWER, AGC, COM, AV:	10.01102448	1.00000000	-1.72924411	0.53862	7
TIME, S/N	DR, POWER, AGC, COM, AV:	9.76864910	1.00000000	3.73758197	-1.67830	5
TIME, S/N	DB, POWER, AGC, COM, AV:	9.90666199	1.00000000	-1.61632633	0.72740	1
TIME, S/N	DR, POWER, AGC, COM, AV:	9.84273911	1.00000000	1.67063034	0.81174	6
TIME, S/N	DB, POWER, AGC, COM, AV:	9.61252880	1.00000000	-1.75099254	0.65192	9
TIME, S/N	DR, POWER, AGC, COM, AV:	9.51148891	1.00000000	1.46720009	0.71419	5
TIME, S/N	DB, POWER, AGC, COM, AV:	9.47694397	1.00000000	-1.30367422	-1.59295	7
TIME, S/N	DR, POWER, AGC, COM, AV:	9.39829159	1.00000000	3.30367422	0.87588	6
TIME, S/N	DB, POWER, AGC, COM, AV:	9.44374943	1.00000000	-1.79285347	0.95099	8
TIME, S/N	DR, POWER, AGC, COM, AV:	9.42841530	1.00000000	-1.90712476	1.08964	6
TIME, S/N	DB, POWER, AGC, COM, AV:	9.47929573	1.00000000	-1.96915901	-1.39335	1
TIME, S/N	DR, POWER, AGC, COM, AV:	9.42067909	1.00000000	3.23909736	-1.32380	9
TIME, S/N	DB, POWER, AGC, COM, AV:	9.45152092	1.00000000	3.31425142	-1.21695	9
TIME, S/N	DR, POWER, AGC, COM, AV:	9.47462940	1.00000000	3.27110457	-1.15740	2
TIME, S/N	DB, POWER, AGC, COM, AV:	9.51918030	1.00000000	-2.06683183	1.45711	0
TIME, S/N	DR, POWER, AGC, COM, AV:	9.43923855	1.00000000	3.15833712	-0.98426	9
TIME, S/N	DB, POWER, AGC, COM, AV:	9.45390797	1.00000000	-2.07930255	1.53248	9
TIME, S/N	DR, POWER, AGC, COM, AV:	9.34943295	1.00000000	-2.14129853	1.47520	3
TIME, S/N	DB, POWER, AGC, COM, AV:	9.30244255	1.00000000	-2.22377682	1.52610	5
TIME, S/N	DR, POWER, AGC, COM, AV:	9.32301521	1.00000000	2.86704087	-0.88815	4
TIME, S/N	DB, POWER, AGC, COM, AV:	9.31636333	1.00000000			

Figure B-1. AMD 2900 Microprocessor Controller Transient

APPENDIX C

LISTING OF AMD 2900 MICROPROCESSOR PROGRAM

In this appendix, we provide a detailed listing of the AMD 2900 microprocessor program. This listing is broken into two parts. The first in Figure C-1 gives the program flow in assembly language while the second in Figure C-2 is the compiled and numerically encoded program and data.


```

#N(N) = 0 MEANS MULT BY 1
M(N) = -1 MEANS MULT BY -1
R0= A(N)
R1=R(N)
R2= M(N)
R3,R4= A'
R5,R6= R'
R9,R10= ALPHA
R11= DELAY POINTER FOR LP FILTERS
R13= B
R14= DELAY LENGTH
R15= OFFSET ADDRESS

DURING INNER LOOP
R0= C(N)
R1= C(N-T1-T2)
R2,R9= COMPUTATION REGISTER
R12=K2

#
O= DZ MAR LDCT(O)
BEGIN1:= CLR(,R15) MDW INCMARI RPCT(BEGIN1) #CLEAR MEMORY #;
      =OZ NIMW MAR
      =LDCT(177773)
      ;
      #PASS 5 COEF. #;
      #WAIT #;
      #GET DATA #;
      #STORE IT #;
      #MOVE PTR AND LOOP #;
      ;
      #FETCH DELAY LENGTH #;
      #WAIT FOR IT TO CLEAR #;
      ;
      #FETCH R FOR LOW PASS #;
      ;
      #WAIT #;
      #FETCH K2 #;
      ;
      ;
      #CALCULATE T1+T2 #;
      ;
      #ACCESS FIRST WEIGHT #;
      #INITIALIZE IT #;
      ;
      #ACCESS K1 #;
      #WAIT FOR A(N) #;
      #FETCH A(N) #;
      #A(N)*K1 #;
      #K1 TO MULY #;
      #ACCESS A(N-T) #;
      #FETCH RESULT #;
      #WAIT FOR R(N) #;
      #FETCH R(N) #;
      #R(N) * K1 #;
      #A(N)*K1 TO MULT #;
      ;
      ;
      #ACCESS K1 #;
      #WAIT FOR A(N) #;
      #FETCH A(N) #;
      #A(N)*K1 #;
      #K1 TO MULY #;
      #ACCESS A(N-T) #;
      #FETCH RESULT #;
      #WAIT FOR R(N) #;
      #FETCH R(N) #;
      #R(N) * K1 #;
      #A(N)*K1 TO MULT #;
      ;

```

Figure C-1. Assembly Language Microprocessor Program

```

=FBA MULY MDR DZ RORS(R0,R2)
=FRA MUMSR MDR RORS(R0,R1) DZ
INP3:= NINT MOV(R2,R0) CJP(INP3)
=RADD(R15) DA DATA=177000 MAR
=MVI(R2) PDP0UT NOSTAT
=RORS(R2,R2) CJS(FIX1)
=RMINS(R15,R14) CAR1
=FQ MULSR DZ NOSTAT
=OUTI(R9) MULX NOSTAT
=MVI(R9) MULY MUMSR MINUS CJP(INIT1)
=CLR(R15) CJP(INIT2)
INIT1:= CONT
INIT2:= MVI(R5) MUMSB
=ZQ FR RORS(R10)
=SZERO FQ2BQ RADD(R5,R6) ZA
=ZQ MULY NOSTAT
=FQ MULSB DZ NOSTAT
=MUMSR DZ RADD(R6) TFR
=SEXT DA RADD(R5,R5) FR
=AO RADD(R6,R6) FB
=RADD(R0,R5) ZB LASTCAR FRA MULX
=OUTI(R0) MULY
=RORS(R2,R2) CJS(FIX2)
=MULSB DA FB RMINS(R10,R4) CAR1
=DMA LPCOEF NOSTAT
=MUMSB DA LASTCAR RMINS(R9,R3) FB CJS(FIX3)
; PRELOOP MULTIPLY FOR FILTER
R5,Q = A*X(N)
=FQ2BQ ZA RADD(R5,R1) SZERO
=ZQ MULX
=MDR MULY DZ
=OUTI(R5) MULX
=TFR RADD(R6) DZ MUMSR
=SEXT MVI(R5)
=ADI(R6,R6) MULSB
=DMA NUMW NOSTAT
=MUMSB RADD(R5,R5) FR DA LASTCAR
=DZ MDR LOSEO

=DMA NTIT2
=RADD(R7) DA CTAB MAR FQ
=RADD DQ MAR MDR
=FRA RORS(R11,R0) DZ MDR MAR
=MVI(R1) MDR
; BEGIN INNER LOOP HERE.
THIS LOOP IS EXECUTED ONCE FOR EACH ANTENNA ELEMENT
CALOOP:= FBA RANDS(R13,R2) ZR INCMARI
=MDR DZ FQ
=FQ2BQ SZERO MDR DZ RADD(R2)
=ZQ MULY
=SEXT MVI(R2)
=DZ MULY MDR
=DZ TFR RORS(R8) MUMSB SZERO
=MULSB ADI(R8,R8)
=MUMSB LASTCAR RADD(R2,R2) FB DA
=RORS(R1,R1)
=MINUS RMA(R11) CJP(NEG)

; A(N)*K1 TO MULT ;
; DELAY A(N) ;
; WAIT FOR M(N) ;
; ACCESS R(N-T) ;
; FETCH M(N) ;
; FIX MULT BY +-1 ;
; TEST FOR WRAPAROUND ;
; A(N)*R(N-T) ;
;
; WRAP AROUND ;
;
; BEGIN DOUBLE PRECISION MULT ;
;
; SET CORRECT SIGN INTO Q ;
; LSH TO MULT ;
; GET LSH OF PREVIOUS MULT ;
; GET FIRST PART OF RESULT ;
; SIGN EXTEND INTO MSB OF RESULT ;
; FINISH LSH OF RESULT ;
; FINISH MSB AND START A*A ;
;
; FIX SIGN AND MULT BY +-1 ;
;
; ACCESS LOW PASS COEFF ;
;
; ATTACH SIGN BIT TO Q ;
; BEGIN MULT MSB ;
; A TO MULT ;
; BEGIN MSB OF MULT ;
; FETCH RESULT FROM LSH ;
; SIGN EXTEND INTO RESULT ;
;
; ACCESS NUMBER OF WEIGHTS ;
; R5,R6 IS RESULT ;
; LOAD LOOP COUNTER ;
;
; ACCESS N-T1-T2 ;
; ACCESS C(N) ;
; ACCESS C(N-T1-T2) ;
; READ C(N) ;
; READ C(N-T1-T2) ;
;
; MULX SR TO MULX ;
; LSH OF Y(N-1) TO Q ;
; SET CORRECT SIGN INTO Q ;
; BEGIN MULT FOR MSB ;
; SIGN EXTEND INTO MSB OF RESULT ;
; MSB TO MULT ;
; LSR*A ;
; MSB*A + LSR*A/2INT. ;
;
; TEST C(N-T1-T2) ;
; RESET MEMORY POINTER ;

```

Figure C-1 continued


```

      =MINUS RMA(R11) CJP(NEG)
      =ADD(R6,R8) MDW
      =RADDS(R5,R2) LASTCAR FB MDW INCMARI SAT CJP(L2) #WRITE Y(N+1) MSH#;
NEG: = SJRA(R6,R8) MDW
      =SMINR(R5,R2) LASTCAR FB MDW INCMARI SAT #WRITE Y(N+1) MSH #;
L2: = ASL(R1,R1)
      =RADDS(R11) DA MAR DATA=177677 NOSTAT #ACCESS WI #;
      =MINUS CJP(L3)
      =SIJAB(R4,R8)
      =RMINS(R3,R2) SAT FB LASTCAR MULX INCMARI CJP(L4) ;
L3: = ADD(R4,R8)
      =RADDS(R3,R2) LASTCAR FB SAT ;
      =ZA RMINS(R8,R8) FB CAR1
      =ZA RMINS(R2,R2) FB LASTCAR MULX INCMARI ;
#
R2,R8 IS DELTA WI
#
L4: = RMINS(R8,R8) MDR DA FB CAR1
      =RMINS(R2,R2) LASTCAR FB MULY SAT MDR DA # WI TO MULT #;
      =MDW RADDS(R2,R2) ZA F2B SARITH
      =RADDS(R11) DA DATA=177677 MAR
      =FRA RADDS(R8,R11) ZB CAR1 MDW
# R2,R8 = WI DELTA WI*WI IN MULT
R12 = K2
#
      =ASL(R0,R0)
      =SEXT RXORS(R12) DA FO NOSTAT
      =CARRY=0 RADDS ZQ FO
      =RADDS(R2,R2) AQ FB SAT POPIN
OUTW:= POPIN SENS CJP(OUTW)
      =FBA RADDS(R2,R11) ZB CAR1 POPIN NOSTAT #TRANSFER #;
      =OUTI(R11) MAR RPCT(CALOOP)
#
END OF INNER LOOP
#
      =DZ RORS(R0) FB MAR ALPHA
      =RMINS(R4,R10) FO CAR1
      =RMINS(R3,R9) FB LASTCAR INCMARO
      =MDR MULY DZ
      =FQ2BQ SZERO RADDS(R9,R10) ZA
      =ZQ MULX
      =OUTI(R9) MULX
      =MUMSB DZ RORS(R10) TFB
      =MVI(R9) SEXT
      =ADI(R10,R10) MDR INCMARO
      =FBA MAR ZB SMINR(R0,R7) NOSTAT
      =ADI(R9,R9) MDR LASTCAR INCMARO SAT ;
      =ADI(R10,R10) MULSB
      =OUTI(R10) MDW NOSTAT
      =RANDS(R7,R7) DA FB DATA=177740 NOSTAT INCMARO #WRAP AROUND C(N)#;
      =ADI(R9,R9) MUMSB LASTCAR MDW SAT CJP(OUTLOOP) #WRITE AND LOOP #;
FIX1: = MDR DZ CAR1 SMINR MULX MINUS CRTN #MULT BY -1 #;
      =MDR DZ MULX CRTN
FIX2: = ZA RMINS(R1,R1) FB CAR1 MDW MINUS CRTN # MULT BY -1 #;
      =ZA RMINS(R1,R1) FB CAR1 MDW CRTN ;
FIX3: = RADDS(R6,R6) FO
      =RADDS(R5,R5) FB LASTCAR SAT ;
      =MOV(R5,R9)
      =ZQ FB RORS(R10) CRTN ;
#RESET MEMORY POINTER #;
#POSITIVE FILTER #;
#NEGATIVE FILTER #;
#TEST R1 AND UPDTF #;
#TEST C(N-T1-T2) #;
#SUBTRACT #;
#ADD #;
#COMPLEMENT #;
#WI - DELTA #;
MDR DA # WI TO MULT #;
#UPDATE WI MSH #;
#RESET POINTER #;
#UPDATE WI LSB #;
#TEST AND UPDATE C(N) #;
#C(N) * K2 #;
#TWO'S COMPLEMENT #;
#OUTPUT IN R2, CHECK SENSE LINE #;
#WAIT FOR POP #;
#TRANSFER #;
#POINT TO Y(N-1) AND LOOP #;
#ACCESS K3 #;
#FINISH ALPHA CALC #;
#;
#FETCH K3 #;
#FIX SIGN IN Q #;
#LS HALF TO MULT #;
#MSB HALF TO MULT #;
#GET LOW ORDER RESULT #;
#SIGN EXTEND RESULT #;
#ALPHA OLD Q ALPHA NEW #;
#DEC R7, ACCESS K3, ACCESS MSH #;
#ADD THE REST OF DELTA ALPHA #;
#UPDATE RESULT #;

```

Figure C-1 continued

F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 0=2,3,0,10,3,1,3,5,3,1,6,4,0,0,0,0
 D 1=2,6,0,10,3,1,0,4,3,1,4,3,4,0,17,177776
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 2=2,1,0,10,3,1,3,5,3,1,6,4,0,0,0,174000
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 3=2,3,0,10,3,1,3,1,3,1,6,4,6,0,0,177773
 D 4=2,14,0,15,3,1,3,1,3,1,6,4,6,0,0,177773
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 5=2,1,1,10,3,0,3,5,3,1,4,4,0,0,0,0
 D 6=2,1,0,10,3,1,3,4,3,1,6,4,3,0,0,0
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 7=2,6,0,10,3,1,3,0,3,1,6,4,6,0,0,177773
 D 10=2,14,0,15,3,1,3,1,3,1,4,3,4,0,7,177767
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 11=2,1,1,10,3,0,3,5,3,1,4,4,0,0,16,0
 D 12=2,1,0,10,3,1,3,1,3,1,6,4,6,0,0,0
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 13=2,14,0,15,3,1,3,1,3,1,4,3,4,0,11,177764
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 14=2,1,1,10,3,0,3,5,3,1,4,4,0,0,15,0
 D 15=2,1,0,10,3,1,3,1,3,1,6,4,6,0,0,0
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 16=2,14,0,15,3,1,3,1,3,1,6,4,6,0,0,177761
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 17=2,1,1,10,3,0,3,5,3,1,4,4,0,0,14,0
 D 20=2,1,0,10,3,1,3,5,3,1,6,4,0,0,0,173776
 D 21=2,1,0,10,3,1,3,1,3,1,6,4,6,0,0,0
 D 22=2,1,0,10,3,1,6,1,3,1,4,7,2,16,0,0
 D 23=2,1,0,10,3,1,3,4,3,1,6,3,2,0,0,177740
 D 24=2,1,0,10,3,1,3,5,3,1,6,4,0,0,0,175576
 D 25=2,1,0,10,3,1,3,4,3,1,6,4,0,0,0,167777
 D 26=2,1,0,10,3,1,3,5,3,1,6,4,0,0,0,173777
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 27=2,14,0,15,3,1,3,1,3,1,6,5,3,11,0,177750
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 30=2,1,1,10,3,0,3,5,3,1,4,4,0,0,0,0
 D 31=2,1,0,10,3,1,3,7,3,1,5,4,0,0,13,175677
 D 32=2,1,0,10,3,1,6,6,3,1,6,4,0,0,0,0
 D 33=2,1,0,10,3,1,3,5,2,1,5,7,3,17,17,0
 D 34=2,1,0,10,3,1,1,1,3,1,4,4,0,0,0,0
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 35=2,14,0,15,3,1,3,1,3,1,4,3,4,0,12,177742
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0

Figure C-2. Numerically Encoded Program and Data

F2 32-30,27-24,23-20,17-0
 D 36=2,1,1,10,3,0,3,5,3,1,4,4,0,0,1,0
 D 37=2,1,0,10,3,1,3,7,3,1,6,4,3,1,0,0
 D 40=2,1,0,10,3,1,6,6,3,1,5,4,0,0,2,0
 D 41=2,1,0,10,3,1,1,4,3,1,5,4,0,0,1,0
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 42=2,14,0,15,3,1,3,1,3,1,4,4,3,2,0,177735
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 43=2,1,0,10,3,1,3,5,3,1,6,7,2,17,0,177000
 D 44=2,1,1,10,3,0,3,5,3,1,4,4,0,0,2,0
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 45=2,16,0,10,3,1,3,1,3,1,6,4,6,2,2,177602
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 46=2,1,0,10,3,1,3,1,2,1,6,5,6,17,16,0
 D 47=2,1,1,10,3,1,2,1,3,1,7,4,0,0,0,0
 D 50=2,1,1,10,3,1,3,7,3,1,6,4,3,11,0,0
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 51=2,14,0,11,3,1,1,6,3,1,4,4,0,0,11,177724
 D 52=2,14,0,10,3,1,3,1,3,1,4,3,4,0,17,177723
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 53=2,1,0,10,3,1,3,1,3,1,6,4,6,0,0,0
 D 54=2,1,0,10,3,1,1,1,3,1,4,4,0,0,5,0
 D 55=2,1,0,10,3,1,3,1,3,1,4,4,5,0,12,0
 D 56=2,1,0,10,3,1,3,1,3,1,3,7,3,5,6,0
 D 57=2,1,1,10,3,1,3,6,3,1,6,4,5,0,0,0
 D 60=2,1,1,10,3,1,2,1,3,1,7,4,0,0,0,0
 D 61=2,1,0,10,3,1,1,1,3,1,0,7,0,0,6,0
 D 62=2,1,0,10,3,1,4,1,3,1,4,7,2,5,5,0
 D 63=2,1,0,10,3,1,3,1,3,1,4,7,7,6,6,0
 D 64=2,1,0,10,3,1,3,7,1,1,5,7,4,0,5,0
 D 65=2,1,0,10,3,1,3,6,3,1,6,4,3,0,0,0
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 66=2,16,0,10,3,1,3,1,3,1,6,4,6,2,2,177600
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 67=2,1,0,10,3,1,2,1,2,1,4,5,2,12,4,0
 D 70=2,1,1,10,3,1,3,5,3,1,6,4,0,0,0,173775
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 71=2,16,0,10,3,1,1,1,1,1,4,5,2,11,3,177576
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 72=2,1,0,10,3,1,3,1,3,1,3,7,3,5,1,0
 D 73=2,1,0,10,3,1,3,7,3,1,6,4,5,0,0,0
 D 74=2,1,0,10,3,1,6,6,3,1,6,4,0,0,0,0
 D 75=2,1,0,10,3,1,3,7,3,1,6,4,3,5,0,0
 D 76=2,1,0,10,3,1,1,1,3,1,0,7,0,0,6,0
 D 77=2,1,0,10,3,1,4,1,3,1,4,4,0,0,5,0
 D 100=2,1,0,10,3,1,2,1,3,1,4,7,2,6,6,0
 D 101=2,1,1,10,3,1,3,5,3,1,6,4,0,0,0,174000

Figure C-2 continued

D 101=2,1,1,10,3,1,3,5,3,1,6,4,0,0,0,174000
 D 102=2,1,0,10,3,1,1,1,1,1,4,7,2,5,5,0
 D 103=2,1,0,10,3,1,6,3,3,1,6,4,0,0,0,0
 D 104=2,1,0,10,3,1,3,5,3,1,6,4,0,0,0,173776
 D 105=2,1,0,10,3,1,3,5,3,1,7,7,2,7,0,167777
 D 106=2,1,0,10,3,1,6,5,3,1,6,7,1,0,0,0
 D 107=2,1,0,10,3,1,6,5,3,1,5,4,0,13,0,0
 D 110=2,1,0,10,3,1,6,1,3,1,4,4,0,0,1,0
 D 111=2,1,0,10,3,1,0,7,3,1,5,3,4,15,2,0
 D 112=2,1,0,10,3,1,6,1,3,1,7,4,0,0,0,0
 D 113=2,1,0,10,3,1,6,1,3,1,3,7,0,0,2,0
 D 114=2,1,0,10,3,1,3,6,3,1,6,4,5,0,0,0
 D 115=2,1,0,10,3,1,4,1,3,1,4,4,0,0,2,0
 D 116=2,1,0,10,3,1,6,6,3,1,6,4,0,0,0,0
 D 117=2,1,0,10,3,1,1,1,3,1,0,4,0,0,10,0
 D 120=2,1,0,10,3,1,2,1,3,1,4,7,2,10,10,0
 D 121=2,1,0,10,3,1,1,1,1,1,4,7,2,2,2,0
 D 122=2,1,0,10,3,1,3,1,3,1,6,4,6,1,1,0
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 123=2,14,0,11,3,1,3,5,3,1,6,4,3,13,0,177651
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 124=2,1,0,10,3,1,3,4,3,1,4,7,6,6,10,0
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 125=2,14,0,10,3,1,0,4,1,0,4,7,6,5,2,177647
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 126=2,1,0,10,3,1,3,4,2,1,4,6,6,6,10,0
 D 127=2,1,0,10,3,1,0,4,1,0,4,6,6,5,2,0
 D 130=2,1,0,10,3,1,3,1,3,1,0,7,3,1,1,0
 D 131=2,1,1,10,3,1,3,5,3,1,6,7,2,13,0,177677
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 132=2,14,0,11,3,1,3,1,3,1,6,4,6,0,0,177642
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 133=2,1,0,10,3,1,3,1,2,1,4,5,6,4,10,0
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 134=2,14,0,10,3,1,0,7,1,0,4,5,6,3,2,177636
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 135=2,1,0,10,3,1,3,1,3,1,4,7,6,4,10,0
 D 136=2,1,0,10,3,1,3,1,1,0,4,7,6,3,2,0
 D 137=2,1,0,10,3,1,3,1,2,1,4,5,3,10,10,0
 D 140=2,1,0,10,3,1,0,7,1,1,4,5,3,2,2,0
 D 141=2,1,0,10,3,1,6,1,2,1,4,5,2,10,10,0
 D 142=2,1,0,10,3,1,6,6,1,0,4,5,2,2,2,0
 D 143=2,1,0,10,2,1,3,4,3,1,2,7,3,2,2,0
 D 144=2,1,0,10,3,1,3,5,3,1,6,7,7,13,0,177677
 D 145=2,1,0,10,3,1,3,4,2,1,5,7,4,10,13,0
 D 146=2,1,0,10,3,1,3,1,3,1,0,7,3,0,0,0
 D 147=2,1,1,10,3,1,4,1,3,1,7,1,2,14,0,0
 D 150=2,1,0,10,3,1,3,1,0,1,7,7,5,0,0,0

Figure C-2 continued

D 150=2,1,0,10,3,1,3,1,0,1,7,7,5,0,0,0
 D 151=2,1,0,10,3,0,7,5,3,0,4,7,7,2,2,0
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 152=2,14,0,06,3,0,7,5,3,1,6,4,6,0,0,177625
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 153=2,1,1,10,3,0,7,5,2,1,5,7,4,2,13,0
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 154=2,6,0,10,3,1,3,5,3,1,6,4,3,13,0,177666
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 155=2,1,0,10,3,1,3,5,3,1,4,4,0,0,0,173774
 D 156=2,1,0,10,3,1,3,1,2,1,7,5,6,4,12,0
 D 157=2,1,0,10,3,1,3,0,1,1,4,5,6,3,11,0
 D 160=2,1,0,10,3,1,6,6,3,1,6,4,0,0,0,0
 D 161=2,1,0,10,3,1,3,1,3,1,3,7,3,11,12,0
 D 162=2,1,0,10,3,1,3,7,3,1,6,4,5,0,0,0
 D 163=2,1,0,10,3,1,3,7,3,1,6,4,3,11,0,0
 D 164=2,1,0,10,3,1,1,1,3,1,0,4,0,0,12,0
 D 165=2,1,0,10,3,1,4,1,3,1,4,4,0,0,11,0
 D 166=2,1,0,10,3,1,6,0,3,1,4,7,2,12,12,0
 D 167=2,1,1,10,3,1,3,5,3,1,5,6,4,0,7,0
 D 170=2,1,0,10,3,1,6,0,1,0,4,7,2,11,11,0
 D 171=2,1,0,10,3,1,2,1,3,1,4,7,2,12,12,0
 D 172=2,1,1,10,3,1,3,4,3,1,6,4,3,12,0,0
 D 173=2,1,1,10,3,1,3,0,3,1,4,3,2,7,7,177740
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 174=2,14,0,10,3,1,1,4,1,0,4,7,2,11,11,177751
 F1 67-66,65-62,61-61,60-55,54-53,52-52,51-47,46-44,43-42,41-41,40-36,35-33
 F2 32-30,27-24,23-20,17-0
 D 175=2,5,0,11,3,1,6,7,2,1,6,6,0,0,0,0
 D 176=2,5,0,10,3,1,6,7,3,1,6,4,0,0,0,0
 D 177=2,5,0,11,3,1,3,4,2,1,4,5,3,1,1,0
 D 200=2,5,0,10,3,1,3,4,2,1,4,5,3,1,1,0
 D 201=2,1,0,10,3,1,3,1,3,1,7,7,6,6,6,0
 D 202=2,1,0,10,3,1,3,1,1,0,4,7,6,5,5,0
 D 203=2,1,0,10,3,1,3,1,3,1,4,4,3,5,11,0
 D 204=2,5,0,10,3,1,3,1,3,1,4,4,5,0,12,0

Figure C-2 continued

